

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**

**DIPLOMOVÁ PRÁCE**

**2017**

**Bc. Roman Heczko**

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

**SENT Bus analyzer na platformě NI RIO v prostředí  
LabVIEW**

SENT Bus analyzer on platform NI RIO in LabVIEW

## Zadání diplomové práce

Student:

**Bc. Roman Heczko**

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2612T041 Řídicí a informační systémy

Téma:

SENT Bus analyzer na platformě NI RIO v prostředí LabVIEW  
SENT Bus Analyzer on Platform NI RIO in LabVIEW

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Seznámení se s problematikou NI RIO a komunikace SENT.
2. Návrh struktury aplikace pro kontrolu signálu sběrnice v časové a napěťové úrovni a logování chybových stavů.
3. Implementace aplikace pro NI RIO v prostředí LabView.
4. Verifikace řešení a testování.
5. Zhodnocení výsledků závěrečné práce a závěr.

Seznam doporučené odborné literatury:

- [1] MARTINEK, Radislav. *Senzory v průmyslové praxi*. 1. vyd. Praha: BEN - technická literatura, 2004. ISBN 80-7300-114-4.
- [2] VLACH, Jaroslav a Josef HAVLÍČEK. *Začínáme s LabVIEW*. 1. vyd. Ilustrace Viktorie Vlachová. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [3] KREIDL, Marcel a Radislav ŠMÍD. *Technická diagnostika*. 1. vyd. Praha: BEN, 2006, 406 s. Technická diagnostika - senzory, metody, analýza signálu. ISBN 80-7300-158-6.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jan Žídek, CSc.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Ing. Jiří Koziolek, Ph.D.  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## **Prohlášení studenta**

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Rád bych poděkoval všem, jejichž rady přispěly ke zpracování této diplomové práce. Zvláště pak děkuji panu doc. Ing. Janu Žídkovi, Csc. za vedení mé diplomové práce a panu Ing. Petru Pětvaldskému za konzultace a poskytnutí odborných rad při jejím vypracování.

V Ostravě, dne 19. 4. 2017

  
.....  
podpis

## Prohlášení zástupce spolupracující právnické osoby

Continental Automotive Czech Republic s.r.o.  
Na Rovince 879  
72000 Ostrava - Hrabová  
Czech Republic

„Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

Z tohoto souhlasu jsou vyjmuty veškeré části bakalářské práce, které obsahují popis produktů, technického řešení, software, hardware a dále pak veškerého know-how a informací, které mohou být takto klasifikovány společností Continental Automotive Czech Republic s.r.o., které nejsou veřejně dostupnými informacemi.

S těmi částmi práce, musí být nakládáno jako s neveřejnou částí práce.



Continental Automotive Czech Republic, s.r.o.  
Kopanská 1713, 744 01 Frenštát p.R.  
IČ 62024922 DIČ CZ62024922 -26-

  
Ing. Petr Pětvaldský

V Ostravě 7.4.2017

Continental Automotive Czech Republic s.r.o.

## **Abstrakt**

Podstatou práce je vytvoření aplikace, pro kontrolu parametrů komunikace SENT v časové a napěťové oblasti a následné logování vzniklých chybových stavů do tabulky v aplikaci a do souboru. Pro získání potřebných informací je nutné seznámit se s normou komunikace SENT a zařízením myRIO – 1900. Získané informace budou použity k vytvoření návrhu řešení a posléze i samotné aplikace. Poté bude vytvořena aplikace otestována dle zadaných podmínek.

## **Abstrakt**

The essence of the work is to create an application to check the SENT communication parameters in time and voltage areas and subsequent logging of error conditions to the table and to a file. To obtain the necessary information, you must familiarize yourself with communication SENT standard and myRIO – 1900 device. The information will be used to create design solution and then the application itself. Created application will be tested according to the specified conditions.

## **Klíčová slova**

SENT, FIFO, časová oblast, napěťová oblast, LabVIEW, myRIO

## **Keywords**

SENT, FIFO, time area, voltage area, LabVIEW, myRIO

## Obsah

Seznam použitých symbolů a zkratk .....	1
Seznam obrázků .....	2
Seznam tabulek .....	3
1. Úvod .....	4
2. NI RIO a komunikace SENT .....	5
2.1 NI MYRIO .....	5
2.1.1 Architektura LabVIEW RIO .....	7
2.1.2 FPGA .....	9
2.2 SENT Bus .....	11
2.2.1 Fyzická vrstva SENT .....	11
2.2.2 Definice jednotlivých zpráv .....	14
3. Návrh struktury aplikace pro kontrolu signálů sběrnice SENT v časové a napěťové úrovni a logování chybových stavů .....	20
3.1 Struktura aplikace .....	20
3.2 Vývojový diagram časové oblasti .....	22
3.3 Vývojový diagram napěťové oblasti .....	23
3.4 Vývojový diagram logování chybových stavů .....	24
4. Implementace aplikace .....	25
4.1 FPGA target .....	25
4.2 Real-Time aplikace .....	27
4.3 Časová oblast .....	29
4.4 Napěťová oblast .....	31
4.5 Logování chybových stavů .....	34
4.6 Dekódování rychlých a pomalých zpráv .....	36
5. Verifikace řešení a testování .....	37
6. Zhodnocení .....	38
Literatura .....	39

## **Seznam použitých symbolů a zkratek**

**FPGA** – Field-Programmable Gate Array

**SENT** – Single Edge Nibble Transmission

**CAN** – Controller Area Network

**LIN** – Local Interconnect Network

**ECU** – Electronic Control Unit (Elektronická řídicí jednotka)

**EMC** – Electromagnetic Compatibility (Elektromagnetická kompatibilita)

**ESD** – Electrostatic Discharge (Elektrostatické vybíjení)

**CRC** – Cyclic Redundancy Check (Cyklický redundantní součet)



## Seznam obrázků

Obr. 1. NI myRIO	3
Obr. 2. Konektory NI myRIO-1900	4
Obr. 3. Architektura LabVIEW RIO	5
Obr. 4. Propojení vrstev architektury LabVIEW	6
Obr. 5. Kategorie požadavků	6
Obr. 6. Blokové schéma FPGA	7
Obr. 7. Různé části FPGA čipu	8
Obr. 8. Parametry SENT přenosového pulzu	11
Obr. 9. Schéma pro dva 12 bitové signály	12
Obr. 10. Vzor nominálního času nibblu	13
Obr. 11. Složení krátké sériové zprávy	14
Obr. 12. Složení rozšířené sériové zprávy	15
Obr. 13. Složení rozšířené sériové zprávy CRC	16
Obr. 14. Návrh struktury aplikace	18
Obr. 15. Vývojový diagram časové oblasti	20
Obr. 16. Vývojový diagram napěťové oblasti	21
Obr. 17. Vývojový diagram logování chybových stavů	22
Obr. 18. Čtení digitálního signálu	23
Obr. 19. Čtení analogového signálu	24
Obr. 20. Synchronizace FIFO	25
Obr. 21. Vyčtení proměnných z fronty	26
Obr. 22. SubVI pro kontrolu Tick time hodin	27
Obr. 23. SubVI pro jitter hodin a chybu driftu	27
Obr. 24. SubVI pro kontrolu přesnosti hodin	28
Obr. 25. Získání hodnot analogového signálu	29
Obr. 26. SubVI pro získání hodnoty napětí	29
Obr. 27. Triggrování subVI	30
Obr. 28. SubVI pro měření délky trvání	31
Obr. 29. Smyčka logování chybových stavů	32
Obr. 30. SubVI pro logování chybových stavů	32
Obr. 31. Dekódování rychlých zpráv	34
Obr. 32. Dekódování pomalých zpráv	35
Obr. 33. Testovací schéma aplikace	37
Obr. 34. Nastavení parametrů komunikace SENT	37
Obr. 35. Zobrazení zpráv a kontrola časové oblasti	38
Obr. 36. Graf napěťového signálu	39
Obr. 37. Zobrazení napěťových úrovní	39
Obr. 38. Tabulka chybových stavů	40

## **Seznam tabulek**

Tab. 1. Komunikační tolerance hodin	11
Tab. 2. Parametry signálu	12
Tab. 3. Popis nibblu stavu a komunikace	16
Tab. 4. Nedetekované 3 a 4 bitové chyby	18

# 1. Úvod

Cílem této diplomové práce je vytvořit analyzátor pro sběrnici SENT na platformě NI RIO v prostředí LabVIEW. Tato aplikace bude kontrolovat signály sběrnice v časové a napěťové oblasti, a logovat chybové stavy.

První část této práce je věnována seznámení se s platformou NI RIO a komunikací pro sběrnici SENT. Jelikož platforma NI RIO obsahuje více zařízení, bylo pro moji práci zvoleno vestavné zařízení NI myRIO – 1900. Toto vestavné zařízení v sobě zahrnuje vše potřebné pro řešení této diplomové práce. Bylo nutné se s platformou NI RIO seznámit a naučit se s ní pracovat, zejména pak s FPGA vrstvou, kterou zvolené vestavné zařízení obsahuje. Dále bylo nutné seznámit se s komunikací po sběrnici SENT a zjistit jakým způsobem pracuje.

Druhá část popisuje návrh aplikace pro kontrolu signálů sběrnice v časové a napěťové oblasti a logování chybových stavů. Při kontrole parametrů časové oblasti vycházíme z digitálního signálu a následných výpočtů, zatímco při kontrole parametrů napěťové oblasti vycházíme z analogového signálu. Logování chybových stavů je prováděno do tabulky umístěné v aplikaci a do souboru na místním uložišti.

Následující část diplomové práce se zabývá implementací aplikace pro platformu NI RIO. Nejprve byly vytvořeny subVI pro kontrolu parametrů časové oblasti. Dalším krokem byla kontrola parametrů napěťové oblasti, a posledním krokem bylo vytvoření části pro logování chybových stavů. Tyto postupy a subVI byly postupně implementovány do aplikace pro čtení parametrů obou signálů.

Čtvrtá část této práce popisuje postup verifikace navrženého řešení a testování výsledků tohoto řešení. Je zde popsáno, kde a jak testování probíhalo, jaké problémy se v jeho průběhu vyskytly a jaké výsledky byly získány.

Poté následuje závěr, ve kterém byl shrnut samotný průběh tvorby aplikace a výsledky získané při testování.

## 2. NI RIO a komunikace SENT

Tato část práce popisuje samotné zařízení myRIO - 1900 jeho architekturu, jednotlivé konektory, vstupy a výstupy. Dále je v této části popsána komunikace SENT podle normy SAE J2716 a obsahuje všechny informace potřebné k dokončení této práce.

### 2.1 NI MYRIO

National Instruments myRIO-1900 je přenosné rekonfigurovatelné I/O (RIO) zařízení, které studenti mohou využít k návrhu řízení, robotiky a mechatronických systémů. NI myRIO-1900 poskytuje analogový vstup (AI), analogový výstup (AO), digitální vstup a výstup (DIO), zvukový a výkonový výstup v kompaktním embedded zařízení. NI myRIO-1900 se připojuje k počítači přes rozhraní USB a bezdrátové rozhraní 802.11b, g, n. [7]



*Obr. 1. NI myRIO. [7]*

Konektory A a B NI myRIO-1900 Expansion Port (MXP) přenášejí totožné sady signálů. Signály se rozlišují softwarově podle názvu konektoru, jako ConnectorA / DIO1 a ConnectorB / DIO1.[7]

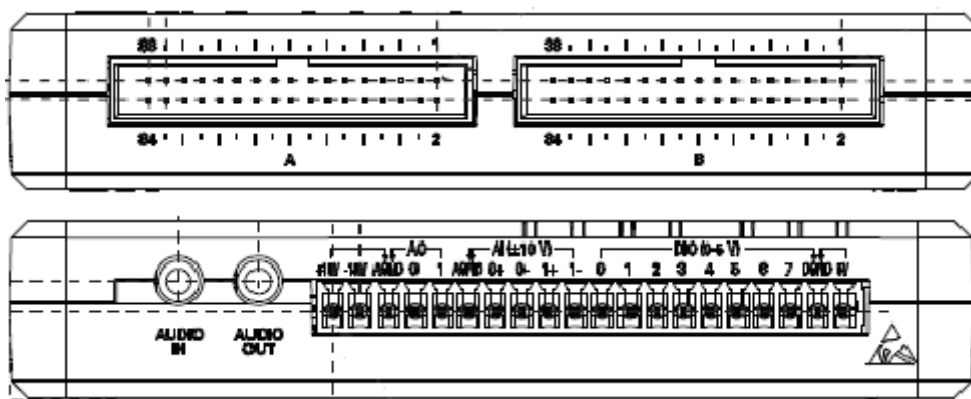
Analogové vstupní kanály NI myRIO-1900 se nacházejí na myRIO Expansion Port (MXP). Analogové signály lze připojit na konektory A, B, Mini systém Port (MSP), konektor C a stereo vstupní konektor audio. Analogové vstupy jsou multiplexovány do jednoho analogově – digitálního převodníku (ADC), který vzorkuje všechny kanály. [7]

MXP konektory A a B mají čtyři analogové vstupy na konektoru AI0 - AI3, které mohou být použity u měření signálů v rozsahu napětí 0 - 5 V. MSP konektor C má dva vysokoimpedanční diferenciální

analogové vstupy AI0 a AI1, které jsou využívány k měření analogových signálů, až po  $\pm 10$  V. Audio vstupy jsou levý a pravý stereo vstup s rozsahem  $\pm 2,5$  V. [7]

Analogové výstupní kanály se rovněž nacházejí na myRIO Expansion Port (MXP) a je možné je připojit na stejné konektory jako kanály vstupní. Každý analogový výstupní kanál má vlastní digitálně-analogový převodník (DAC), takže mohou být všechny kanály aktualizovány současně. Převodníky pro analogové výstupní kanály jsou řízeny dvěma sériovými komunikačními sběrnicemi z FPGA. MXP konektory A a B sdílí jednu sběrnici. MSP konektor C a audio výstupy sdílejí druhou sběrnici. [7]

MXP konektory A a B mají dva analogové výstupní kanály na konektoru AO0 a AO1, které mohou být použity ke generování signálů v rozsahu 0 - 5 V. MSP konektor C má dva analogové výstupní kanály AO0 a AO1, které mohou být využity ke generování signálů do  $\pm 10$  V. Dále myRIO – 1900 obsahuje audio výstupy, což jsou levý a pravý stereo výstup na které lze připojit sluchátka. [7]



Obr. 2. Konektory NI myRIO-1900.

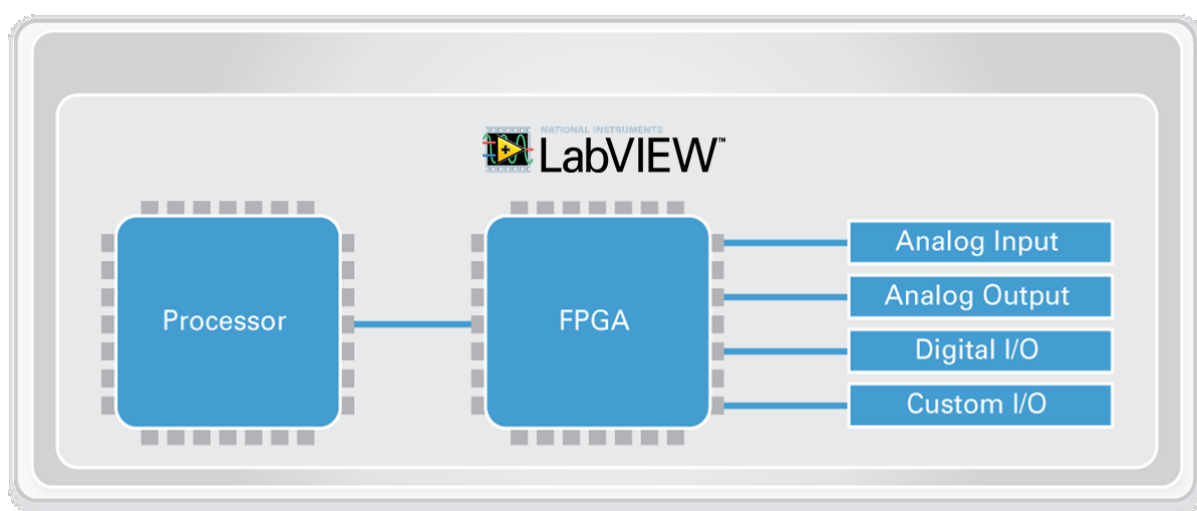
NI myRIO-1900 má 3,3 V univerzální DIO linky na konektorech MXP a MSP. MXP konektory A a B mají 16 DIO linek na konektoru. Na MXP konektorech, každá DIO linka od 0 do 13, má 40 k $\Omega$  pullup rezistor na 3,3 V, a DIO linie 14 a 15 mají 2.2 k $\Omega$  pullup rezistory na 3,3 V. MSP konektor C má osm DIO linek. Každá MSP DIO linka má 40 k $\Omega$  pulldown odpor k zemi. DGND je společný GND pro všechny DIO linky. Naprogramovat lze všechny linky jednotlivě jako vstupy nebo výstupy. Sekundární digitální funkce zahrnují Serial Peripheral Interface (SPI), I2C, pulzní šířkovou modulaci (PWM) a kvadrurní enkodér.[7]

Na zařízení NI myRIO-1900 je rovněž několik LED diod pro signalizaci. První z nich je LED dioda „Power“ signalizující, že má zařízení připojené napájení. Další je LED dioda „Status“, která je vypnutá během běžného provozu. Zařízení provádí samočinný test (POST) po přivedení napájení. V průběhu tohoto testu diody „Power“ a „Status“ svítí a po jeho ukončení dioda „Status“ zhasne, což signalizuje, že test byl dokončen. Pokud během testu nastala chyba, bude dioda „Status“ blikat. Zařízení dále obsahuje čtyři programovatelné LED diody, které je možné využít k signalizaci průběhu programu. [7]

### 2.1.1 Architektura LabVIEW RIO

Architektura LabVIEW RIO kombinuje Real-Time procesory, uživatelsky programovatelné FPGA, a modulární I/O s jednotným softwarovým prostředím, které mohou být použity k programování a přizpůsobení každého prvku systému. Se schopností propojení s jakýmkoliv typem snímače, nebo speciálních I/O, pokročilým zpracováním signálu a vlastní analýzou, a prováděním pokročilých řídicích algoritmů, je architektura LabVIEW RIO ideální pro navrhování pokročilých řídicích nebo monitorovacích systémů. [5]

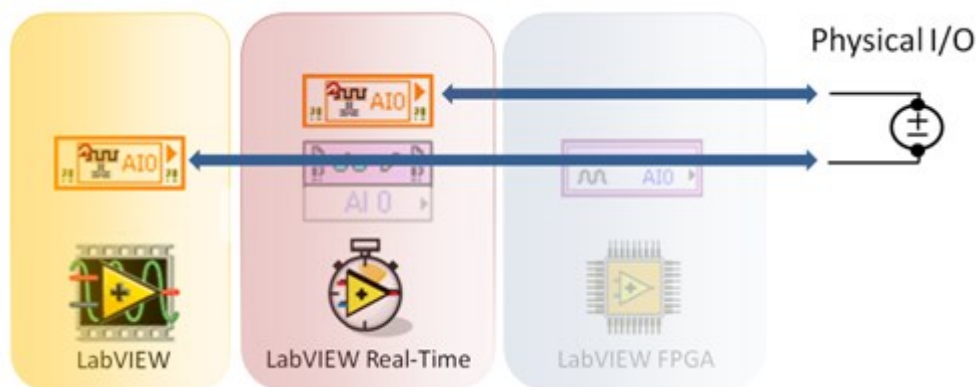
Architektura LabVIEW RIO integruje čtyři složky: Real-Time procesor, uživatelsky programovatelné FPGA, modulární I/O, a kompletní softwarový řetězec nástrojů pro naprogramování každé aplikace z hlediska architektury. [5]



*Obr. 3. Architektura LabVIEW RIO. [5]*

Rekonfigurovatelné FPGA (Field-Programmable Gate Array) je jádrem architektury LabVIEW RIO a může být použito k přesunutí kritických nebo náročných úkolů z procesoru na tuto část architektury, která poskytuje spolehlivé, deterministické provedení kódu s extrémně vysokou propustností. FPGA je připojeno přímo k I/O pro vysoce výkonné zpracování signálu a obrazu a nastavitelné časování, spouštění a synchronizaci.

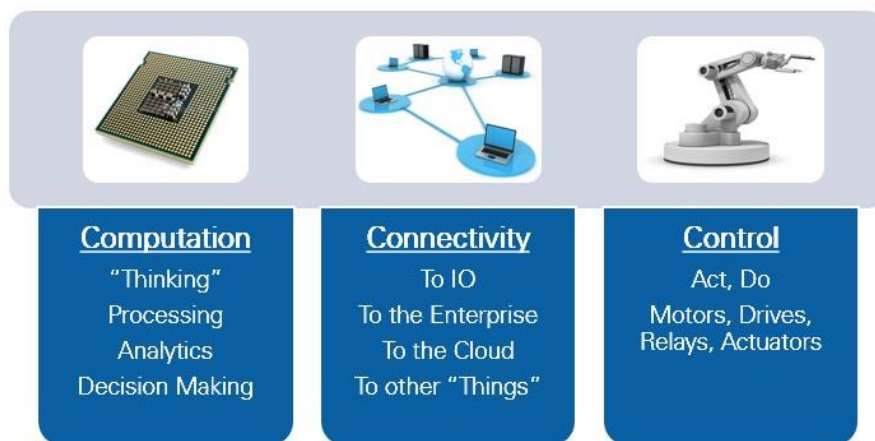
Při přímém připojení FPGA k I/O je prakticky nulová latence regulační smyčky, proto výkon tohoto zařízení postačuje pro většinu pokročilých algoritmů řízení. NI hardware na základě architektury LabVIEW RIO využívá nejnovější technologie FPGA a čipové sady od Xilinx, od serie Artix až po Virte, a to až do 410T, která zahrnuje 406,720 logických buněk, 1,540 množství řezu DSP, a 28MB bloku RAM.



Obr. 4. Propojení vrstev architektury LabVIEW.[8]

Standardní softwarové prostředí obsahuje kompletní sadu vestavěných matematických a analytických funkcí, algoritmy pro zpracování signálů a obrazů, sítě a I/O API rozhraním. Na NI Linux Real-Time Modulu je open-source Real-Time operační systém založený na Linuxu vytvořeném firmou Angström Distribution, který běží na embedded procesoru a přináší spolehlivé, deterministické operace a kompletní sadu rozhraní API pro ukládání dat, přenos dat a vlastní zpracování a analýzu dat. LabVIEW FPGA modul umožňuje programovat embedded FPGA v intuitivním grafickém programovacím prostředí, bez znalosti jazyků pro popis technického vybavení jako VHDL nebo Verilog. [5]

Při provádění kontrolních a monitorovacích aplikací musí návrháři zvážit vše od, synchronizace až po časování. Všechny požadavky lze shrnout do tří kategorií: výpočty, připojení a ovládání. [5]



Obr. 5 Kategorie požadavků. [5]

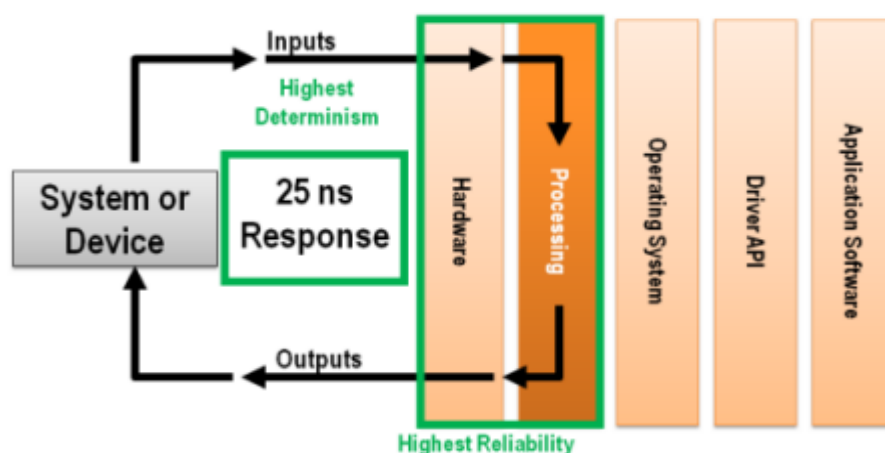
Při výpočtech může systém provádět řadu algoritmů zpracování a analýzy, které se pohybují od booleovské logiky po složitou matematiku. V reálných aplikacích musí být digitální systémy propojeny s fyzickým světem prostřednictvím senzorů, také s jinými digitálními systémy, jako jsou podnikové sítě, databáze, cloud-based služby, a dokonce i jiné stroje, zařízení a infrastruktury. Pro ovládání mají embedded systémy výstupy, které jsou generovány v důsledku vnitřních algoritmů a analýz. Tyto výstupy mohou ovládat motory, relé, nebo akční členy. [5]

Architektura LabVIEW RIO je ideální pro systémy, které vyžadují výpočty, konektivitu a ovládání. Modulární I/O nabízí přizpůsobitelné rozhraní schopné uspokojit prakticky jakýkoli požadavek na připojení například analogové a digitální senzory, průmyslové komunikační protokoly, vlastní I/O, a dokonce i speciální I/O, jako jsou kamery a motory. Uživatelsky programovatelné procesory a FPGA mohou být využity k provedení pokročilých zpracování obrazu nebo signálů, provádění složitých výpočtů a provádění rozhodovacích algoritmů. [5]

FPGA je zejména ideální pro řízení, protože je rychlé, deterministické a pracuje paralelně. Dokáže provést desítky regulačních smyček paralelně, každou s jedinečným časováním a mechanismy pro přenos dat. [5]

### 2.1.2 FPGA

Programovatelné hradlové pole (FPGA) jsou reprogramovatelné křemíkové čipy. Ross Freeman, spoluzakladatel Xilinx, vynalezl první FPGA v roce 1985. Přijetí FPGA čipů ve všech průmyslových odvětvích je dáno tím, že FPGA čipy kombinují nejlepší části integrovaných obvodů (ASIC) a procesorově založených systémů. FPGA poskytují hardwarově časovanou rychlost a spolehlivost. [6]



Obr. 6. Blokové schéma FPGA. [6]

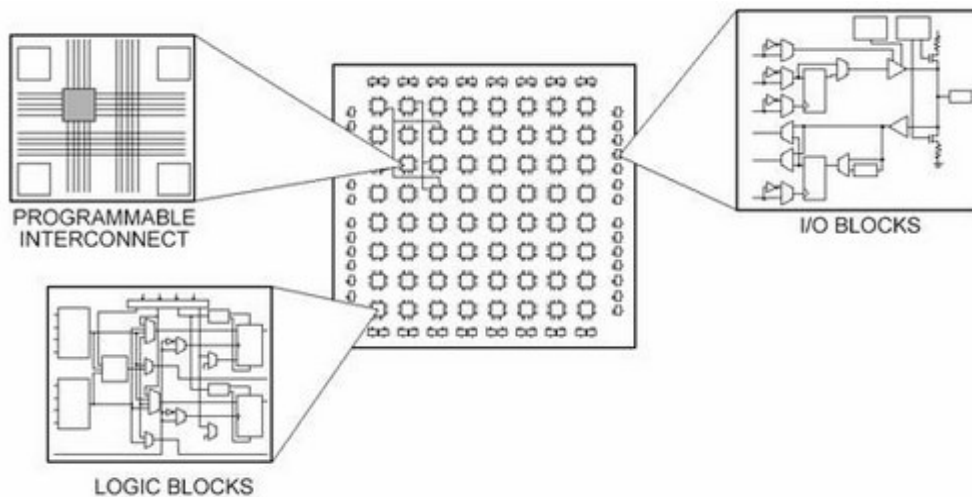
Jednou z výhod FPGA na rozdíl oproti systémům na bázi procesorů je, že aplikační logika není prováděna operačním systémem, ovladači a aplikačním software, ale v hardwarových obvodech. [6]

Reprogramovatelné pole hradel má obdobnou flexibilitu softwaru jako software běžící na procesorově založeném systému, ale jeho běh není omezený počtem jader, která jsou k dispozici. Na rozdíl od procesorů, FPGA pracují skutečně paralelně, takže různé procesy nemusejí soupeřit o stejné zdroje. Každý nezávislý úkol zpracování je přiřazen k vyhrazené části čipu, a může fungovat autonomně bez vlivu jiných logických bloků. Výsledkem je, že výkon jedné části aplikace není ovlivněn, když je přidáno další zpracování. [6]

Každý čip FPGA je vyroben s konečným počtem předdefinovaných zdrojů s programovatelným propojením rekonfigurovatelných číslicových obvodů a vstupně - výstupních bloků. [6]



Specifikace zdrojů FPGA často obsahují řadu konfigurovatelných logických bloků, počet logických bloků jako multiplikátory, velikost paměti zdrojů paměti jako jsou vložené bloky paměti RAM. Jsou to právě tyto zdroje, které jsou nejvíce důležité při výběru FPGA pro konkrétní aplikaci. [6]



*Obr. 7. Různé části FPGA čipu. [6]*

Konfigurovatelné logické bloky (CLB) jsou základní logickou jednotkou FPGA. Někdy označovány jako řezy nebo logické buňky, CLB se skládají ze dvou základních částí: „flip-flops“ a „lookup tables“ (LUT). Různé FPGA rodiny se liší způsobem, jakým jsou flip-flopy a LUT zabaleny dohromady. [6]

## 2.2 SENT Bus

Komunikace SENT je určena pro použití v aplikacích, kde jsou data snímače s vysokým rozlišením přenášena ze senzoru do řídicí jednotky motoru. Tento systém je zamýšlený jako náhrada za metody přenosu s menším rozlišením a jako jednodušší nízkonákladová alternativa k CAN nebo LIN sběrnici. Při implementaci se předpokládá, že senzor obsahuje mikroprocesor nebo vyhrazený logický obvod pro vytvoření signálu. [4]

SENT je jednosměrný komunikační systém. Směr vysílání je od senzoru (vysílací zařízení) k řídicí jednotce (přijímací zařízení), která neobsahuje synchronizační signály. Signál senzoru je přenášen jako série pulzů. Data jsou zakódována jako periody signálu od sestupné k sestupné hraně. [4]

### 2.2.1 Fyzická vrstva SENT

Fyzická vrstva je zodpovědná za poskytnutí způsobu přenášení digitálních dat zakódovaných jako doba mezi dvěma sestupnými hranami signálu přes komunikační médium. To má za následek opakovatelné a přesné časové spínání vstupního obvodu rozhraní fyzické vrstvy, která se skládá z napájecích, zemnicích a signálových vodičů. [4]

#### Normální režim komunikace a přenosová rychlost

Přenosová bitová rychlost je variabilní v závislosti na periodě taktu hodin, odeslané datové hodnotě a nominální hodnotě odchylky času taktu hodin a +20% tolerance hodin. Nejdelší přenosový čas pro 6 datových nibblů je 270 taktů hodin nebo 972  $\mu$ s, při 3  $\mu$ s periodě taktu hodin. Takto získáme nejhorší přenosovou rychlost ~24.7 kBits/s. Podobně, nejkratší platná zpráva o 6 datových nibblech je 154 taktů hodin s -20% tolerancí taktu hodin. Výsledná přenosová rychlost bude ~64.9 kBits/s. Tyto kalkulace ignorují změny, které nastávají při použití sériového přenosu dat. [4]

#### Komunikace vysílač – přijímač

Tabulka číslo 1 zobrazuje požadavky na časování a rozlišení pro vysílací a přijímací moduly. Variace taktů hodin vysílače je velká, aby nevyžadoval vysoce nákladný oscilátor. Požadavky na chybu Driftu hodin a jitter při středních periodách taktu hodin, musí být lineárně interpolované mezi minimem a maximem period taktů hodin. [4]

Typ zařízení	Parametr	Tolerance hodin
Vysílač	Přesnost hodin	$\leq \pm 20\%$ na čas taktu hodin bez pause pulzu $> -20\%$ , $< +18\%$ , na čas taktu hodin s pause pulzem
Vysílač	Jitter hodin a chyba Driftu	$\leq 0.3 \mu$ s variace maximálního času nibblu v porovnání s očekávaným časem odvozeným z kalibračního času pulzu při 3 $\mu$ s taktu hodin
Vysílač	Perioda taktu hodin	3 do 90 $\mu$ s nominální
Přijímač	Jitter hodin a chyba Driftu	$\leq 0.3 \mu$ s variace maximálního času nibblu v porovnání s očekávaným časem odvozeným z kalibračního času pulzu při 3 $\mu$ s taktu hodin

Tab. 1. Komunikační tolerance hodin. [4]

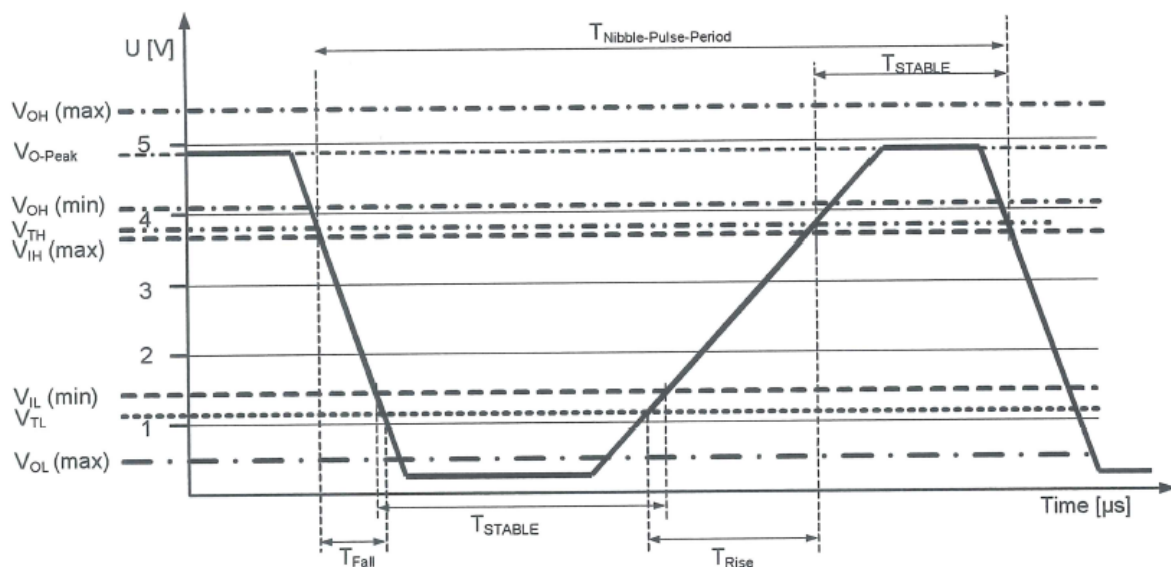
### Požadavky elektrického rozhraní

SENT signál může být považován za nominální 5 V obdélníkový signál, ale pro nízké vyzařované rušení je zapotřebí tvarování signálu. Tento signál se používá v automobilovém průmyslu pro systémové aplikace „pod kapotou“. Vysílač a přijímač musí být chráněny před zkraty na kostru nebo napětí vozidla a různými přechodovými pulzy jak je uvedeno níže a ve specifických požadavcích na EMC vozidla. Pro minimalizaci efektu offsetu pro zem a napájení, musí přijímací zařízení přivádět regulovanou 5 V referenci do přenosového zařízení. Přenosové zařízení může implementovat 5 V regulátor pro napájení signálového driveru nezávisle na napájení přivedeného z přijímacího zařízení. [4]

Parametr			Limity (včetně tolerance komponent)		Jednotky	Testovací podmínky / Definice
			Min	Max		
a.	$V_{OL}$	=		0.5	V	Nízký stav napětí @ 0.1 mA DC zatížení proudu
b.	$V_{OH}$	=	4.1		V	Vysoký stav napětí @ 0.1 mA DC zatížení proudu
c.	$I_{SUP}$	=		50	mA	Průměrný odběr z napájení 5V přijímače přes jednu správu
d.	$I_{SUP-RIPPLE}$	=		9.0	mA	Peak-to-peak rozdíl v spotřebě napájecího proudu přes jednu správu při frekvencích 30 kHz
e1.	$T_{FALL}$	=		6.5(1)	$\mu s$	Od $V_{TH} = 3.8$ V do $V_{TL} = 1.1$ V, $I_{SUP} \leq 20$ mA
e2.	$T_{FALL}$	=		5.0(1)	$\mu s$	Od $V_{TH} = 3.8$ V do $V_{TL} = 1.1$ V, $20$ mA $< I_{SUP} \leq 50$ mA
f.	$T_{RISE}$	=		18(1)	$\mu s$	Od $V_{TL} = 1.1$ V do $V_{TH} = 3.8$ V
g.	$\Delta T_{FALL}$	=		0.1(1)	$\mu s$	Edge to edge jitter ve statickém prostředí pro jakoukoliv periodu pulsu
h.	$T_{STABLE}$	=	6(2)		$\mu s$	Stabilizační čas signálu pod $V_{IL}$ (nízký signál) nebo nad $V_{IH}$ (vysoký signál)

Tab. 2. Parametry signálu. [4]

Přenosové zařízení musí obsahovat driver s konfigurací a technologií vhodnou pro automobilový průmysl. Driver musí zajistit, že parametry pulzu signálu, tak jak jsou definovány, jsou splněny při odesílání na přijímač. Vysílač musí být chráněn před poškozením napětím nebo proudem na výstupním signálu. Jakákoliv situace, která by mohla vést k poškození vysílače, musí vypnout jeho driver, aby omezila protékající proud, a zabránila poškození. Dále je doporučen EMC filtr, skládající se ze sériového zapojení kondenzátoru a rezistoru s výstupním pinem, aby zmínil RF energie vázané na externí signálové vedení. Za účelem splnění požadavků na vyzařované EMC emise se očekává, že „tvarovaná“ křivka bude řízená vysílačem. [4]



Obr. 8. Parametry SENT přenosového pulzu. [4]

Přenosové zařízení může volitelně implementovat 5 V regulátor pro napájení driveru SENT signálu nezávisle na napájení z přijímače. Zatímco 5 V napájení může být nezávislé na přijímači, vysílač musí mít připojený svůj signál zpětného vedení k zemnicímu signálu, aby zajistil, že nadměrný offset zemnění nebude rušit komunikaci. [4]

#### ESD přechodový tlumič

Pokud je to nutné, mohou být přidány obvodové prvky jako Zenerovy diody (back-to-back) nebo varistor na jednom nebo více místech pro poskytnutí ESD ochrany. Nicméně pokud jsou tyto prvky použity, mohou přidat kapacitanci nebo zavést napěťovou anebo teplotní variabilitu pro signálovou časovou konstantu. [4]

#### Kabelový svazek sběrnice a ECU konektory

Odpor všech konektorů musí mít méně než  $1 \Omega$  na vedení za celou délku života vozidla. Na elektroinstalaci sběrnice musí být použity kabely, které mají méně než  $0.1 \text{ nF}$  na metr délky vodiče. Maximální délka kabelu je 5 metrů. [4]

#### ESD imunita

Vstupně výstupní piny vysílacího a přijímacího modulu musí odolat elektrostatickému výboji bez jakéhokoli poškození vysílacího nebo přijímacího modulu, v případě testování pomocí elektrostatického výboje výrobcem vozidla. [4]

#### Požadavky na zkoušku EMC

Fyzická vrstva, pokud je začleněna do ECU nebo designu senzoru, musí fungovat tak, jak je specifikováno pro zařízení zamýšlené pro práci v elektromagnetickém prostředí. Dále, elektromagnetické emise vznikající při operacích souvisejících se signálem, nesmí zasahovat do normální funkce jiných ECU nebo subsystému. [4]

## Režimy ochrany před poruchou

Datová komunikace může být přerušena při zkratu na napájení nebo zemi.

1. Zkrat elektroinstalace k zemi – Nesmí vzniknout poškození žádného zařízení, pokud je signálové nebo napájecí vedení zkratováno k zemi. Impedance menší než 50 Ohmů mezi vedením a zemí musí být považována za zkrat k zemi. Po odstranění této chyby může nastat restart vysílače a normální provoz musí pokračovat v rámci restartovacího času vysílače. Nesmí být potřeba zásahu obsluhy. [4]
2. Zkrat elektroinstalace k napájení – Nesmí vzniknout poškození žádného zařízení, pokud je signálové nebo napájecí vedení zkratováno k napájení senzoru  $V_{OUT}$ . Impedance menší než 50 Ohmů mezi vedením a  $V_{OUT}$  musí být považována za zkrat. Po odstranění této chyby může nastat restart vysílače a normální provoz musí pokračovat v rámci restartovacího času vysílače. Nesmí být potřeba zásahu obsluhy. [4]

## 2.2.2 Definice jednotlivých zpráv

### Obecné požadavky

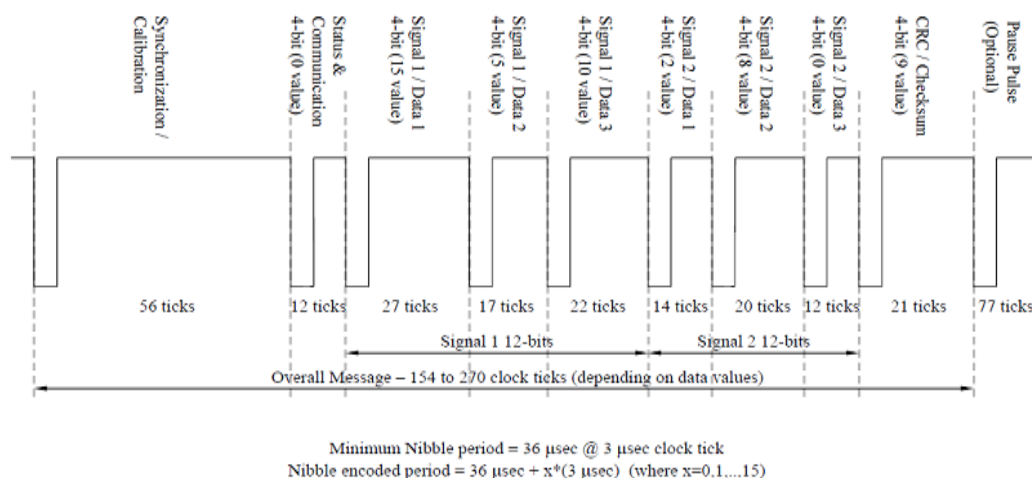
Vysílání nastane nezávisle na jakékoli akci přijímacího modulu, to znamená, že vysílání nesmí požadovat synchronizační signál z přijímače. [4]

### Přenosová sekvence

Kódovací schéma se skládá ze sekvence pulzů, které jsou opakovaně posílány vysílacím modulem. Přenos se skládá z kalibračního (synchronizačního) pulzu s periodou 56 taktů hodin. Poté následuje jeden pulz stavu a sériové komunikace, který má od 12 po 27 taktů hodin. [4]

Dále je odeslána sekvence jednoho až šesti datových nibble pulzů (od 12 po 27 taktů hodin), které představují hodnoty signálů, které mají být přeneseny. Počet nibblů bude pevně daný pro každou aplikaci schéma kódování, ale může se lišit mezi aplikacemi. Například, jestliže jsou přenášeny dvě 12 bitové hodnoty, je v tomto případě odesláno 6 nibblů. [4]

Předposlední částí této sekvence je jeden, 4 bitový Checksum(kontrolní součet) nibble pulz, který má opět od 12 po 27 taktů hodin, a jako poslední je možné odeslat jeden volitelný pause pulz. [4]



Obr. 9. Schéma pro dva 12 bitové signály. [4]

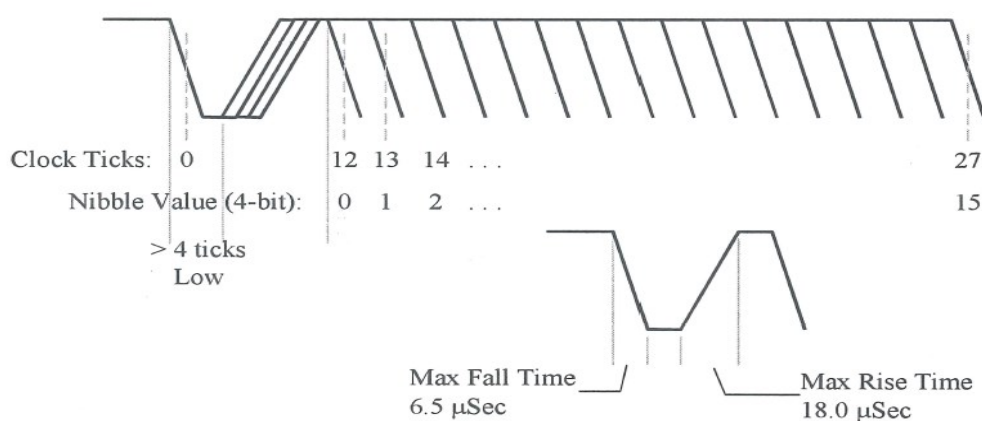
Obrázek číslo 9 zobrazuje přenos jedné zprávy pro dvě 12 bitové senzorové hodnoty při délce taktu hodin 3  $\mu$ s. Jak je zřejmé z obrázku, nejkratší zpráva se skládá ze 154 a nejdelší správa z 270 taktů hodin. Signály předány pomocí datových nibblů jsou rovněž nazývaný jako „Fast Channel“ signály pro tuto specifikaci. [4]

Přenosové vlastnosti kalibračního (synchronizačního) pulzu jsou:

- Nominální perioda pulzu je 56 taktů hodin.
- První 4 takty hodin musí přenášet nulovou hodnotu (zbylé takty mají hodnotu 1).
- Skutečná perioda měřená přijímacím modulem k nápravě variace hodin vysílače. [4]

Přenosové vlastnosti nibble pulzu

- Minimální perioda pulzu je 12 taktů hodin.
- První 4 takty hodin musí přenášet nulovou hodnotu (zbylé takty mají hodnotu 1).
- Každý nibble je 1 takt hodin (0 – 15 nibblů je 0 – 45  $\mu$ s při 3  $\mu$ s taktu hodin)
- Minimální perioda nibble pulzu (přenosová hodnota 0) = 12 taktů hodin (36  $\mu$ s při 3  $\mu$ s taktu hodin).
- Maximální perioda nibble pulzu (přenosová hodnota 15) = 12 + 15 = 27 taktů hodin (36 + 45 = 81  $\mu$ s při 3  $\mu$ s taktu hodin). [4]



Obr. 10. Vzor nominálního času nibblu. [4]

### Nibble stavu a komunikace

Tento nibble je rezervován pro povolení přenosu různých informací jako například části čísel nebo kódu chyby informace. Tento nibble není zahrnutý ve formulaci CRC kalkulace a proto mohou mít větší výskyt nedetekovaných chyb. [4]

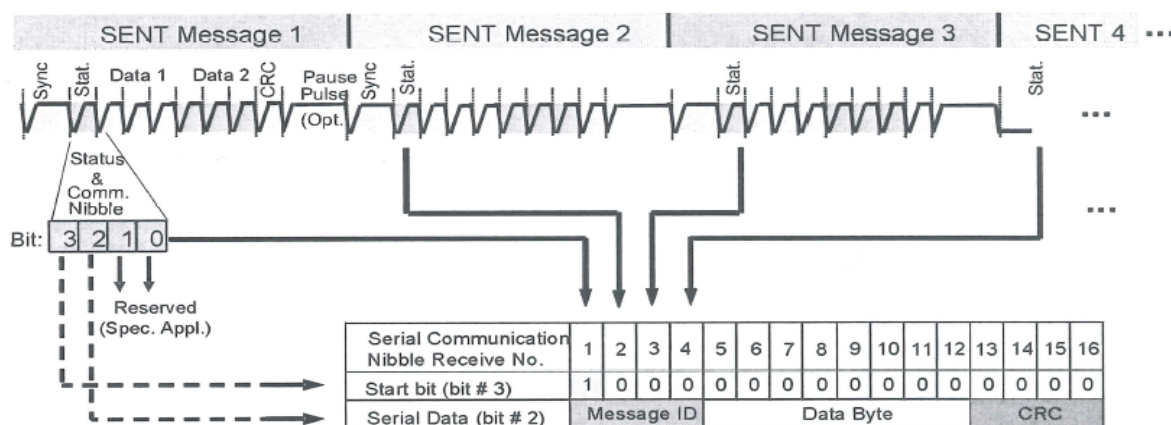
Číslo bitu	Funkce bitu
0 (LSB)	Vyhrazená pro konkrétní aplikaci.
1	Vyhrazená pro konkrétní aplikaci.
2	Bity sériové datové zprávy.
3 (MSB)	Zpráva start = 1, jinak = 0 nebo bity sériové datové zprávy.

Tab. 3. Popis nibblu stavu a komunikace. [4]

Bity 2 a 3 definují volitelný kanál sériové zprávy vysílače, který může být realizován buď ve formátu krátké sériové zprávy anebo ve formátu rozšířené sériové zprávy. SENT přijímače musí podporovat oba tyto formáty. Signály předány pomocí bitů sériové zprávy jsou také známy jako „Slow Channel“ signály pro tuto specifikaci. [4]

### Formát krátké sériové zprávy

Sériová data jsou (bit po bitu) ve druhé bitu po sobě následujících zpráv z vysílače. Sériová data jsou posílána v 16 bitové sekvenci. Počáteční bit sériové zprávy je indikován „1“ na bitu 3 nibblu stavu a komunikace. Dalších 15 přijatých rámců musí obsahovat hodnotu „0“ v té samé pozici na bitu 3. Všechny 16 rámců musí být úspěšně přijato, aby mohla být přijata sériová hodnota. [4]



Obr. 11. Složení krátké sériové zprávy. [4]

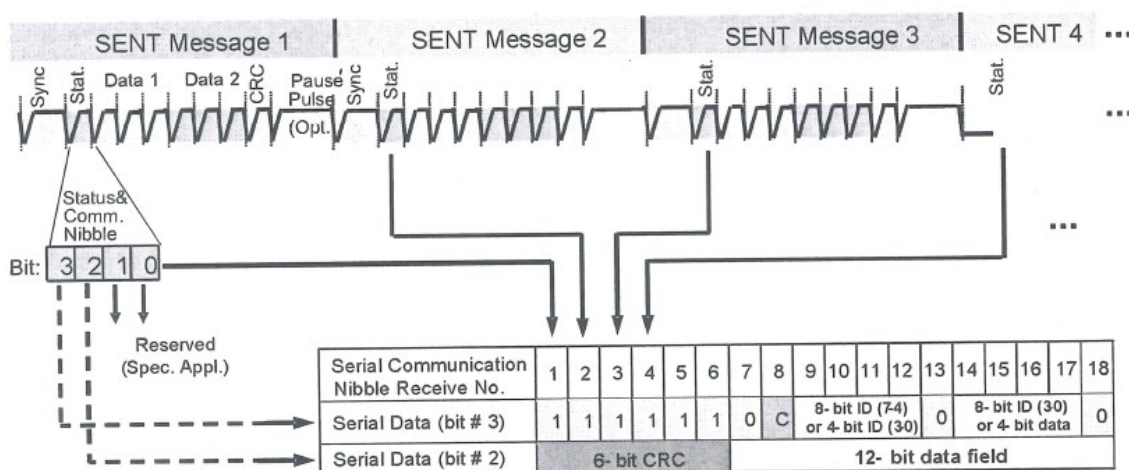
16 bitová zpráva se skládá ze 4 bitového ID nibblu zprávy, 2 datových nibblu (1 byte dat) a CRC checksum nibblu. CRC checksum je odvozený pro ID zprávy a 2 datové nibbly a používá stejný

algoritmus, který je použit ke kalkulaci SENT CRC nibblu. ID zprávy se používá na identifikaci typu přenášených dat. ID a hodnoty dat jsou specifické pro každou aplikaci. Všechna odesílaná data jsou řazena v pořadí od nejvýznamnějšího bitu po nejméně významný bit. [4]

### Formát rozšířené sériové zprávy

Volitelný rozšířený formát sériové zprávy může být použit senzory, které potřebují sériový komunikační kanál s větším datovým polem a větší sadou ID zpráv. Pokud je použitý kanál rozšířené sériové zprávy, potom jsou sériová data přenášena v bitu #2 a bitu #3 nibblu stavu a komunikace. Rámec sériové zprávy se táhne přes 18 po sobě jdoucích datových zpráv SENT z vysílače. Všechny 18 rámců musí být úspěšně přijato, aby mohla být přijata sériová hodnota. [4]

Začátek rámce je indikován jedinečným vzorem "01111110" v bitu 3 nibblu stavu a komunikace. První "1" v sérii šesti jedniček (po "0") indikuje první nibble rámce sériové zprávy. Sériová data bit #3 sériových komunikačních nibblů 1 až 6 je nastaven do "1". Sériová data bit #3 sériových komunikačních nibblů 7, 13 a 18 je nastaven do "0". Při inicializaci je doporučeno, aby vysílač poslal sériovou data bit #3 sekvenci "0...01111110". Jedna počáteční "0" zajistí start sekvence s minimální latencí a proto se žádná přerušovaná zpráva nebude zdát dokončená a nesprávně projde CRC. [4]



Obr. 12. Složení rozšířené sériové zprávy. [4]

Rámec sériové zprávy obsahuje 20 bitů dat. Je možné vybrat si z dvou konfigurací stanovených konfiguračním bitem:

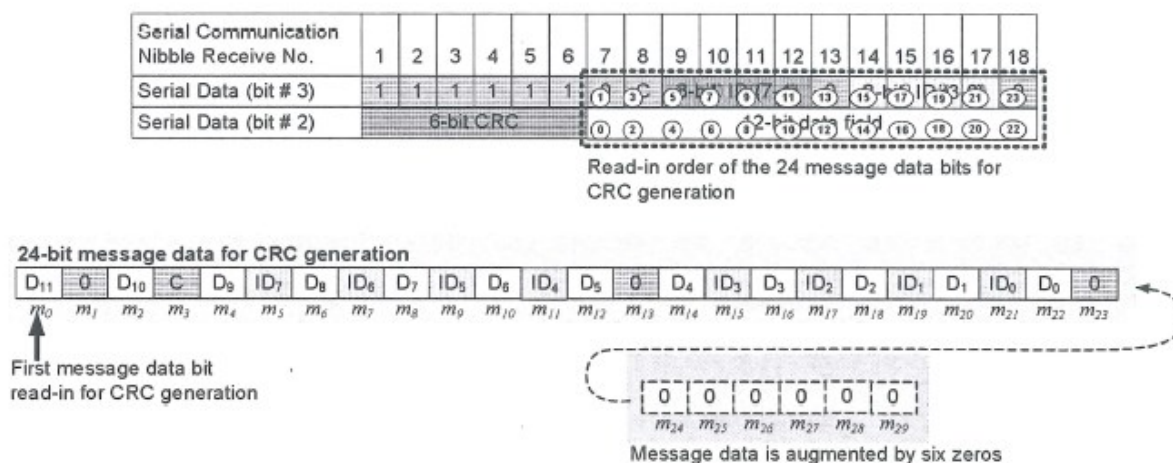
- 12 - bit data a 8 - bit ID zprávy (konfigurační bit = 0)
- 16 - bit data a 4 - bit ID zprávy (konfigurační bit = 1)

Všechna data, která jsou přenášena kanálem sériové zprávy, jsou odesílána v pořadí od nejvýznamnějšího bitu po nejméně významný bit. [4]



## Formát rozšířené sériové zprávy CRC

Tato CRC hodnota je vypočítávána jako funkce obsahu bitů 2 a 3 sériové datové zprávy pro rámec 7-18 (12 bitové datové pole, 8 bitová ID zpráva, konfigurační bit a nulové bity 7, 13 a 18). Za účelem CRC výpočtu musí být bity seřazeny:  $m = [\text{Rámec 7 bit \#2}, \text{Rámec 7 bit \#3}, \text{Rámec 8 bit \#2}, \text{Rámec 8 bit \#3} \dots \text{Rámec 18 bit \#2}, \text{Rámec 18 bit \#3}]$ . Tyto bity se podílejí na datech zprávy  $m = [m_0 m_1 m_2 \dots m_{21} m_{22} m_{23}]$ . Bit  $m_0$ , který je shodný s bitem  $D_{11}$  rámce je první přijatý bit. [4]



Obr. 13. Složení rozšířené sériové zprávy CRC. [4]

Kódování je definováno generujícím polynomem:  $G(x) = x^6 + x^4 + x^3 + 1$  s hodnotou seedu 010101. Pro generaci CRC bude zpráva rozšířena o šest nul  $m_{\text{crc}} = [m_0 m_1 m_2 \dots m_{21} m_{22} m_{23} 0 0 0 0 0 0]$  a 6 bitová data pro výpočet CRC vypadají následovně:

- data (1) =  $\{m_0 (=MSB), m_1, m_2, m_3, m_4, m_5 (=LSB)\}$
- data (2) =  $\{m_6 (=MSB), m_7, m_8, m_9, m_{10}, m_{11} (=LSB)\}$  [4]

Šesti bitový polynom CRC umožňuje detekovat všechny 1 bitové, 2 bitové, 3 bitové a 5 bitové chyby. Množství nedetekovaných 4 bitových chyb závisí na počtu datových bitů zprávy. [4]

Data zprávy	Počet nedetekovaných 3 bitových chyb	Počet nedetekovaných 4 bitových chyb
24 bit	0	365
24 bit + detekování chyb bitů 7, 13 a 18	0	208

Tab. 4. Nedetekované 3 a 4 bitové chyby. [4]

U tří sériových datových bitů #3 rámce číslo 7, 13 a 18 je známo, že jsou nulové. Pokud je kterýkoliv z těchto tří bitů na daných rámcích jiný než nulový je tento rámec chybný. Toto zmenšuje počet nedetekovaných 4 bitových chyb z 365 na 208. [4]

Pokud není přenášena žádná sériová zpráva, musí být oba sériové datové bity nastaveny na nulu v obou formátech. Pokud musí vysílač přerušit již započatou sériovou datovou sekvenci, je povinností vysílače zajistit, aby sériová zpráva, která bude přijata, měla chybný kontrolní součet, což lze způsobit poškozením buďto dat nebo přeneseného CRC. Tento požadavek nezahrnuje restartování vysílače. [4]

### **Pause pulz**

Přenos sběrnice SENT může volitelně také obsahovat pulz navíc, který bude přenesen po nibblu kontrolního součtu. Tento pulz může být například použit k vytvoření SENT přenosu s konstantním počtem taktů hodin. [4]

- Minimální délka – 12 taktů (ekvivalent nibblu s hodnotou 0)
- Maximální délka – 768 taktů ( $3 * 256$ ) [4]

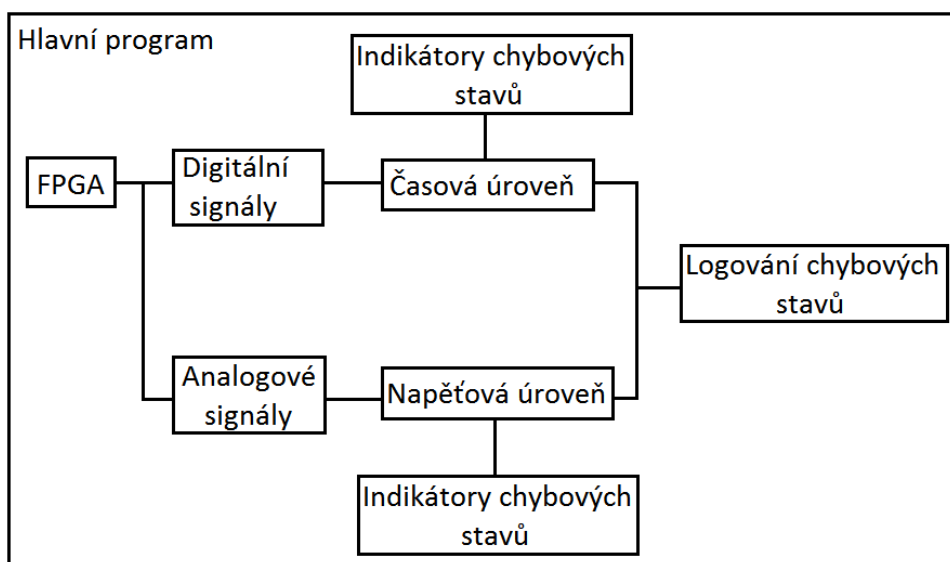
Pro zprávu obsahující 6 datových nibblů by měla být minimální konstantní délka SENT správy 282 taktů pro pause pulz v rozsahu od 12 do 128 taktů. [4]

### 3. Návrh struktury aplikace pro kontrolu signálů sběrnice SENT v časové a napět'ové úrovni a logování chybových stavů

Tato část se zabývá návrhem aplikace pro kontrolu signálů sběrnice SENT v časové a napět'ové úrovni a logováním chybových stavů, které se vyskytnou během komunikace. Na základě získaných informací byla navržena struktura odpovídajících funkcí a provedena jejich následná implementaci. Aplikace byla vytvořena a testována na platformě NI myRIO – 1900.

#### 3.1 Struktura aplikace

Zdrojem informace pro tuto aplikaci je senzor, který přes sběrnici SENT posílá zprávy. Tyto zprávy jsou dále pomocí platformy NI myRIO – 1900, která v sobě zahrnuje FPGA procesor a program LabVIEW dekodována do podoby dat. SENT sběrnice využívá technického standardu J2716.



Obr. 14. Návrh struktury aplikace.

Nejprve je třeba získat potřebná data ze sběrnice SENT. Toto bude provedeno pomocí FPGA vrstvy přítomné na platformě NI myRIO – 1900. Již v této fázi bylo zřejmé, že bude potřeba jak analogový tak digitální signál. Z digitálního signálu budeme dále získávat a kontrolovat časové parametry sběrnice a z analogového napět'ové parametry sběrnice. Real-Time vrstva bude využívána pro kontrolu jednotlivých časových a napět'ových parametrů.

Základním časovým parametrem je perioda taktu hodin, která musí být v mezi 3  $\mu$ s a 90  $\mu$ s. Toto je velmi důležitý parametr, jelikož na něm přímo závisí přenášený signál, protože se délky jednotlivých pulzu uvádějí v počtech taktů hodin.

Dalšími parametry jsou požadavky na jitter hodin a chyba driftu, kde tyto chyby nesmí být  $\leq 0.3 \mu$ s variace maximálního času nibblu v porovnání s očekávaným časem odvozeným z kalibračního času pulzu při 3  $\mu$ s taktu hodin.

Posledním časovým parametrem je přesnost hodin, která by měl být  $\leq \pm 20\%$  na čas taktu hodin bez pause pulzu nebo  $> -20\%$ ,  $< +18\%$ , na čas taktu hodin s pause pulzem. Pro variantu s pause pulzem je tento parametr volitelný.

Napěťové parametry lze posoudit podle tvaru přenosového pulzu SENT sběrnice. Prvním parametrem je nízký stav napětí, pro který je maximální hodnota  $V_{OL} = 0.5 \text{ V}$  a dalším je  $V_{OH} = 4.1 \text{ V}$  což je vysoký stav napětí.

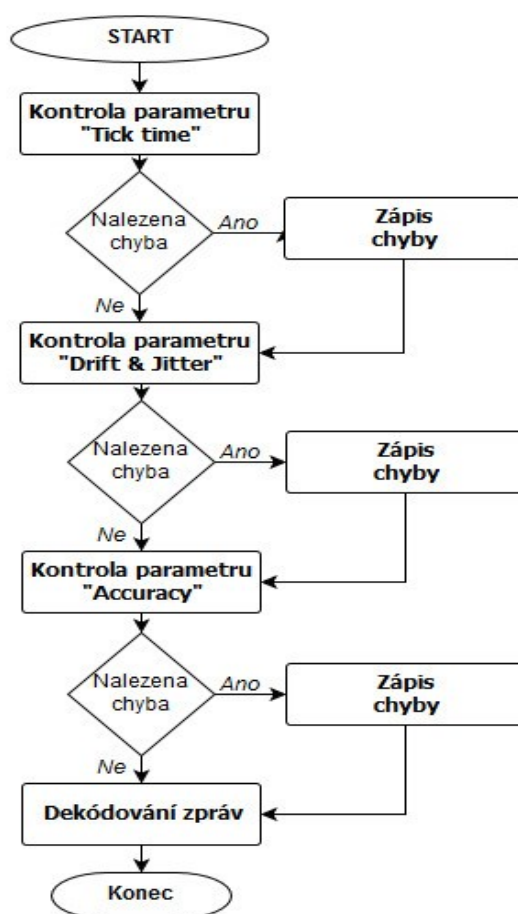
Další součástí programu budou indikátory pro zobrazení hodnot měřených veličin a diody pro zobrazení, jestli daná hodnota nepřesahuje nastavenou mez. Časová úroveň bude obsahovat pouze indikátory a diody. Napěťová úroveň bude obsahovat indikátory, diody a navíc graf pro zobrazení průběhu přenosového pulzu SENT sběrnice.

Poslední částí programu bude logování chybových stavů. Tato část bude postupně zapisovat všechny chybné stavy nejprve do tabulky v aplikaci a posléze do dokumentu.

## 3.2 Vývojový diagram časové oblasti

Tato část popisuje průběh kontroly parametrů časové oblasti a dekodování chybových stavů pomocí k tomu určených subVI. Jednotlivé úkony jsou popsány pomocí vývojového diagramu, který popisuje sled událostí probíhajících v této části aplikace.

Nejprve je spuštěna část zabývající se získáním digitálního signálu. Z tohoto signálu jsou poté získávány jednotlivé proměnné sloužící ke kontrole jednotlivých parametrů. Prvním parametrem je „Tick time“. Pokud je tento parametr v pořádku, pokračuje aplikace ke kontrole následujícího parametru. Pokud není v pořádku, tak je proveden zápis chyby a až poté program pokračuje dále.



Obr. 15. Vývojový diagram časové oblasti.

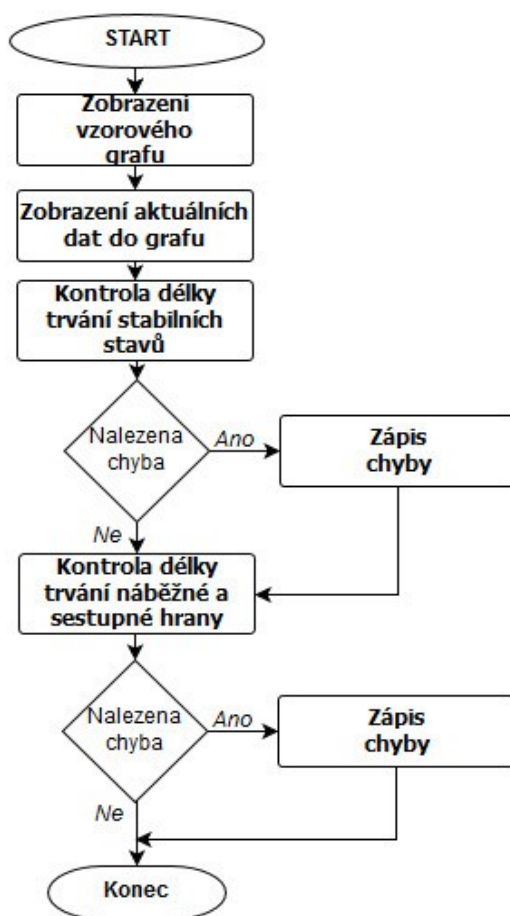
Tento postup se opakuje i pro parametr „Drift & Jitter“ popisující jitter hodin a chybu driftu, a také parametr „Accuracy“ popisující přesnost hodin. Po kontrole všech parametrů je provedeno dekodování rychlých a pomalých zpráv. Tento postup je probíhá cyklicky.

### 3.3 Vývojový diagram napět'ové oblasti

Tato část popisuje průběh kontroly parametrů napět'ové oblasti. Jednotlivé úkony jsou popsány pomocí vývojového diagramu, který popisuje sled událostí probíhajících v této části aplikace.

Jako první je do připraveného grafu vyobrazen vzorový průběh napět'ového signálu. Tento průběh je dán normou SAE J2716 a popisuje jednotlivé napět'ové vrstvy signálu. Dalším krokem je zobrazení získaného napět'ového signálu do druhého grafu. Tento graf bude zobrazovat aktuální průběh signálu.

Vizuální kontrola pomocí zobrazení průběhu v grafu je v tomto případě nutná. Je nutné se ujistit, že přijímaný analogový signál odpovídá svým tvarem průběhu signálu tak, jak jej popisuje norma.



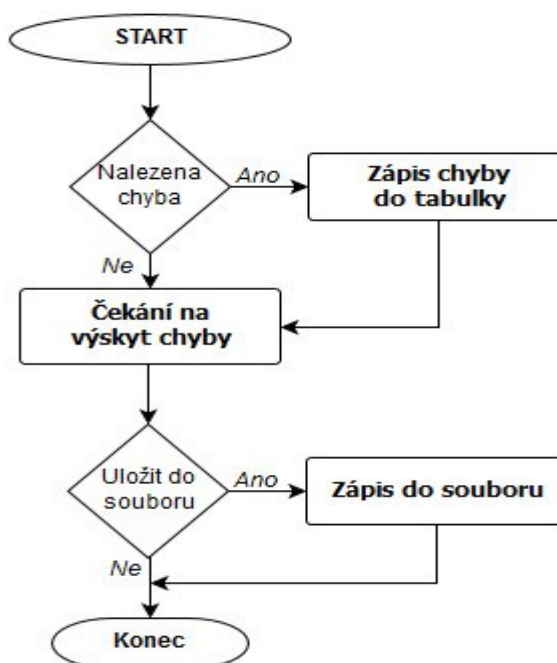
Obr. 16. Vývojový diagram napět'ové oblasti.

Poté je na řadě kontrola délky trvání jednotlivých stabilních stavů a kontrola délky trvání náběžné a sestupné hrany signálu. Opět pokud pro tyto parametry budou nalezeny chyby, zapíší se tyto chyby do připravené tabulky a aplikace bude pokračovat dále v činnosti. Tato část aplikace také probíhá cyklicky.

### 3.4 Vývojový diagram logování chybových stavů

Tato část popisuje průběh logování chybových stavů. Opět je tato část popsána pomocí vývojového diagramu. Po spuštění této části aplikace vždy zkontroluje, jestli má na některém ze vstupů hlášenou chybu. Pokud je na vstupu skutečně nahlášená chyba, je tato chyba zapsána do tabulky spolu s datem a časem výskytu.

Pokud nebyla detekována žádná chyba, tak aplikace čeká na výskyt další chyby. Pokud uživatel aplikace zvolí možnost pro uložení dat z již zapsaných v tabulce do souboru, je proveden tento zápis do předem určeného souboru. Tato část taktéž probíhá cyklicky.



Obr. 17. Vývojový diagram logování chybových stavů.

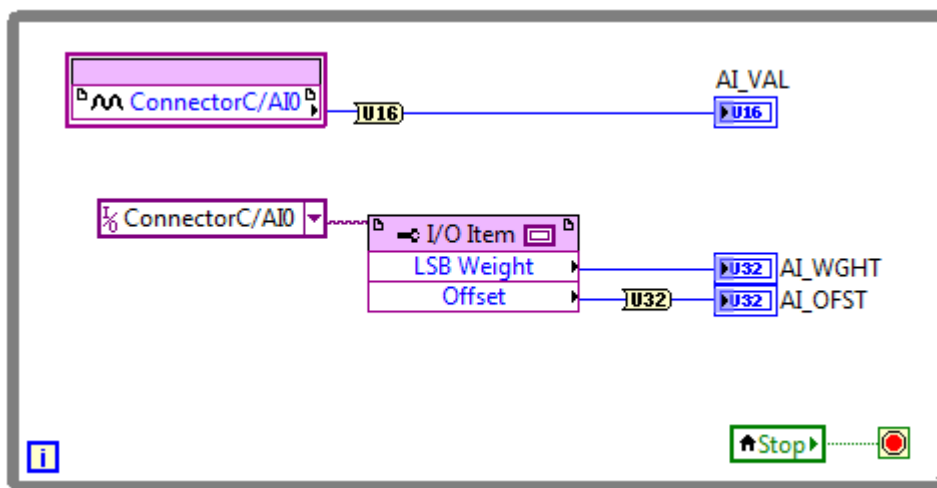




Po přečtení informací z tohoto konektoru jsou data přivedena do subVI pro detekci pulzů. Zde se nejprve detekuje hrana signálu. Dále toto subVI určuje stav, ve kterém se signál nachází a také délku pulzu odvozenou od 40 megahertzového krystalu v FPGA.

Dalším krokem je přenos detekovaných pulzů do Real-Time vrstvy pomocí synchronizační metody FIFO. Celý název této metody je „First In, First Out“, což znamená, že první zaslaný prvek bude vždy prvním vyčteným prvkem. Hodnoty pulzů jsou přivedeny na vstup „Element“ a na vstup „Timeout“ je přivedena nulová konstanta. K výstupu „Timed Out?“ je připojena zpětnovazební smyčka zajišťující zobrazení případného timeoutu na čelním panelu a také tlačítko pro resetování tohoto timeoutu.

Digitální signál je také veden na led diodu dvě, která zobrazuje příchozí signál. Také je zde zajištěno blikání diody číslo jedna v případě, že je program spuštěn a rozsvícení diody číslo tři, když je FPGA vrstva propojená s Real-Time vrstvou.



Obr. 19. Čtení analogového signálu.

Analogový signál získáváme poměrně jednodušším způsobem. Podobně jako u digitálního signálu jsou data vedena přes „FPGA I/O Node“ na který je připojen analogový vstupní port 0, ale poté jsou data pouze konvertována na datový typ U16 a přivedena na proměnnou, která je poté vyčítána v Real-Time vrstvě.

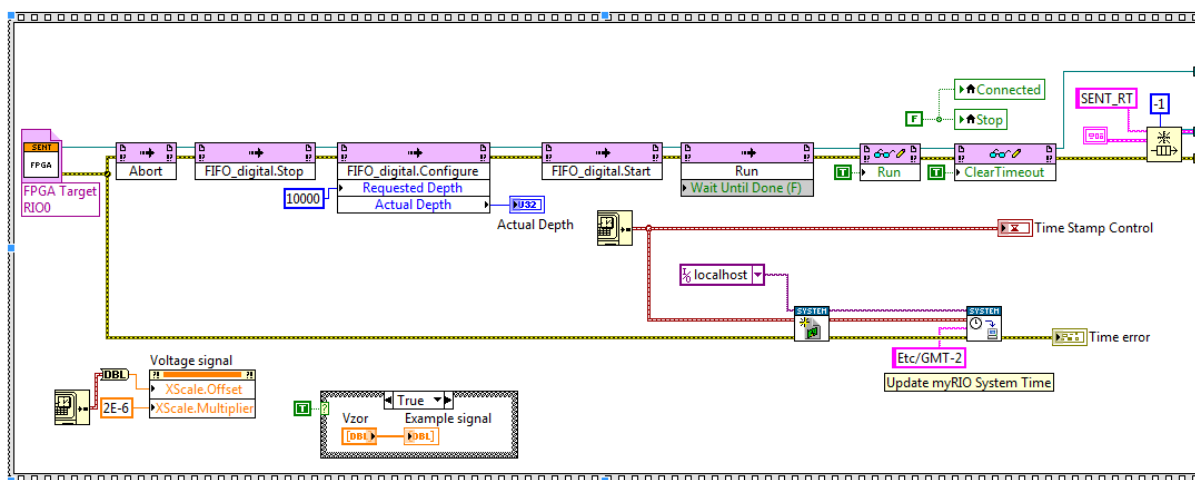
Je potřeba v této části také získávat hodnotu váhy LSB a hodnotu Offsetu, což se provádí pomocí „FPGA I/O Property Node“. Tyto hodnoty jsou uloženy v hardware. Na vstupu této funkce je opět definován konektor a port, pro který chceme tyto hodnoty získat. Po jejich získání jsou tyto hodnoty opět zapsány na proměnné, které jsou poté přenášeny do Real-Time vrstvy. Hodnota Offsetu musí být pro přehlednost převedena na datový typ U32.

Poslední částí sekvence je opět časovaná smyčka se standartními hodinami, která provádí restartování všech led na myRIO-1900 tím že na ně přivede nulovou hodnotu.

## 4.2 Real-Time aplikace

Nejprve je nutné, aby data získána FPGA targetem, byla doručena do Real-Time části programu. Pro tento úkol jsou použity funkce „Invoke method“ ze sekce „FPGA Interface“. Jako první se provede „Abort“ což provádění FPGA zastaví. Dále je vyslán příkaz „FIFO stop“, aby se synchronizace zastavila a poté pomocí metody „Configure“ správně nakonfigurovala na požadovanou hloubku. V tomto případě je hloubka nastavená na desetkrát větší než je požadované číslo elementů.

Poté je zadán příkaz pro spuštění synchronizace a také spuštění přenosu dat. Celá tato část je v „Flat sequence“ což zajišťuje, že se provede jako první. Dále je v této části také inicializována fronta pro pozdější přenos získaných dat do smyčky určené pro jejich zpracování.



Obr. 20. Synchronizace FIFO.

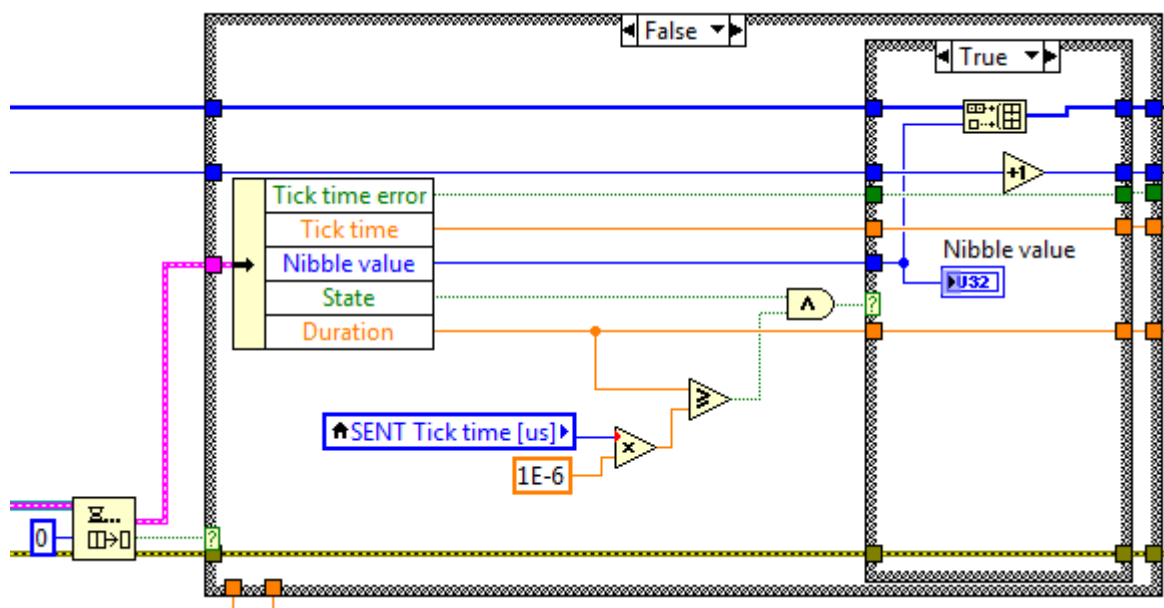
Také je zde nastavení časové osy a offsetu pro graf napětíové oblasti, zobrazení základních úrovní tohoto grafu dle normy SAE J2716 a také přenastavení data a času v samotném zařízení myRIO-1900.

Protože toto zařízení neobsahuje baterii a jeho čas a datum jsou po odpojení resetována do továrního nastavení. K tomuto účelu slouží subVI „Inicitalize“ a „Set time“. Je zde třeba napojit funkci „Get date/time in seconds“ a také při inicializaci specifikovat cíl kde se má datum a čas přenastavit. Poté je také nutné nastavit správně časovou zónu.

Po vytvoření fronty přichází na řadu „While“ smyčka která zajišťuje čtení data synchronizace FIFO, což se provádí pomocí metody „FIFO.Read“. Dalším krokem je zápis do fronty pro další použití.

Další částí je smyčka „For“, která obsahuje dekodování jednotlivých „nibblů“ (pulzů) a také měření jednotlivých taktů hodin. Získaná data se poté pomocí funkce „Bundle by name“ zapisují do fronty, která získaná data posílá do další smyčky „While“. Zde je také připojeno tlačítko „Connected“ pro spuštění přenosu dat dále do programu. Také tato smyčka zobrazuje počet zbývajících elementů a maximální počet zbývajících elementů v synchronizační metodě FIFO. Na konci této smyčky je také řešeno nouzové vypínání programu v případě přetečení, timeoutu nebo zmáčknutí STOP tlačítka.

Pomocí fronty s názvem SENT\_RT jsou data přenášena do další smyčky „While“. Pokud tato fronta nemá timeout, tak se zde vyčtou hodnoty potřebné pro kontrolu parametrů časové oblasti, což jsou „Tick time“, „Duration“, „Tick time error“ a „State“, a také hodnota jednotlivých nibblů. Hodnoty jednotlivých nibblů se zapisují do pole, které používá shift registry pro zapamatování jednotlivých hodnot. Zde je také připojené subVI pro dekódování rychlých zpráv, detekci chyby driftu a jitteru a také chyby přesnosti.

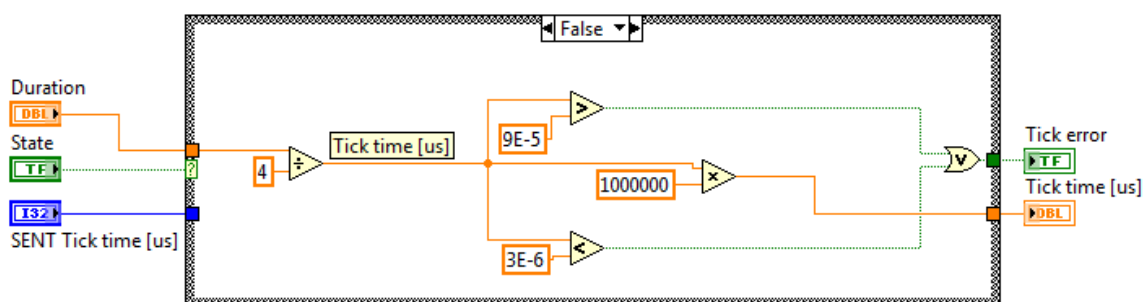


Obr. 21. Vyčtení proměnných z fronty.

Po dekódování rychlé zprávy následuje, „Case“ struktura, která počítá a zobrazuje rychlé a pomalé zprávy. Také je zde řešeno získání pomalé zprávy z výstupu stavu a komunikace subVI pro dekódování rychlých zpráv. Zobrazení pomalých zpráv je opět v „Case“ struktuře, takže je zpráva zobrazena, pouze pokud byla dekódována celá.

### 4.3 Časová oblast

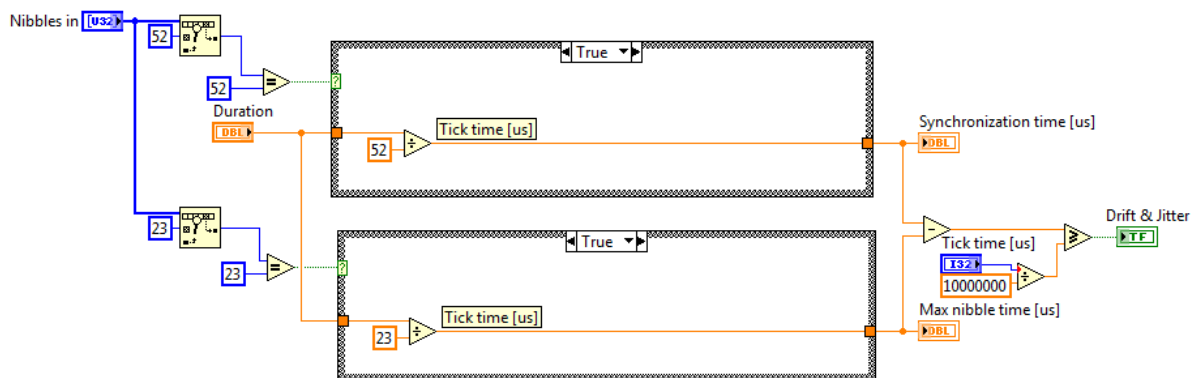
Prvním časovým parametrem je perioda taktu hodin. Tato perioda se musí nacházet v rozmezí 3 až 90  $\mu\text{s}$ . Jelikož se takt hodin vypočítává z délky trvání nulového stavu tak se struktura „Case“ spustí jen v tomto případě. Nejprve je nutné vypočítat pomocí délky trvání nibblu čas taktu a poté jej porovnat, zdali je v daném rozmezí. Čas taktu vypočteme tak, že hodnotu délky trvání podělíme čtyřmi, což je počet taktů nulového stavu. Pokud není v daném rozmezí, tak se rozsvítí signalizační LED dioda. Dále pokud je stav digitálního signálu jedna, tak je na indikátor „Tick time“ přivedena nastavovaná hodnota „SENT Tick time“.



Obr. 22. SubVI pro kontrolu Tick time hodin.

Toto subVI se nachází ve smyčce pro získání digitálního signálu, kde je prováděno pomocí smyčky „While“. Data zde získána jsou poté převáděna do další smyčky pomocí fronty s názvem SENT\_RT.

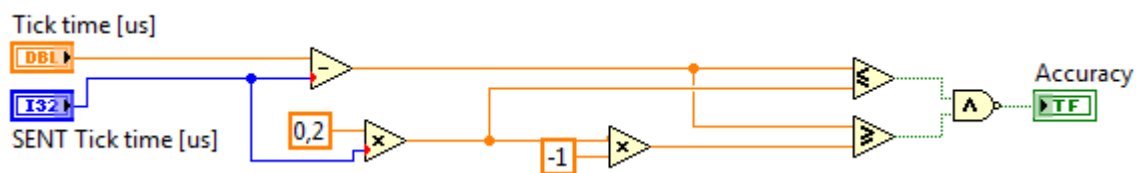
Dalším parametrem je jitter hodin a chyba driftu. V tomto případě je nejprve třeba najít synchronizační pulz o délce 52 taktů hodin a maximální nibbl o délce 23 taktů hodin. Proto je jedním ze vstupů tohoto subVI pole hodnot „Nibbles in“. Pro oba tyto pulzy se pomocí délky trvání nibblu (Duration) vypočte „Tick time“. Tyto časy se poté odečtou. Výsledná hodnota je porovnávána s nastavenou hodnotou „SENT Tick time“.



Obr. 23. SubVI pro jitter hodin a chybu driftu.

Toto subVI se nachází ve smyčce pro dekódování rychlých a pomalých zpráv, kde je prováděno pomocí smyčky „While“.

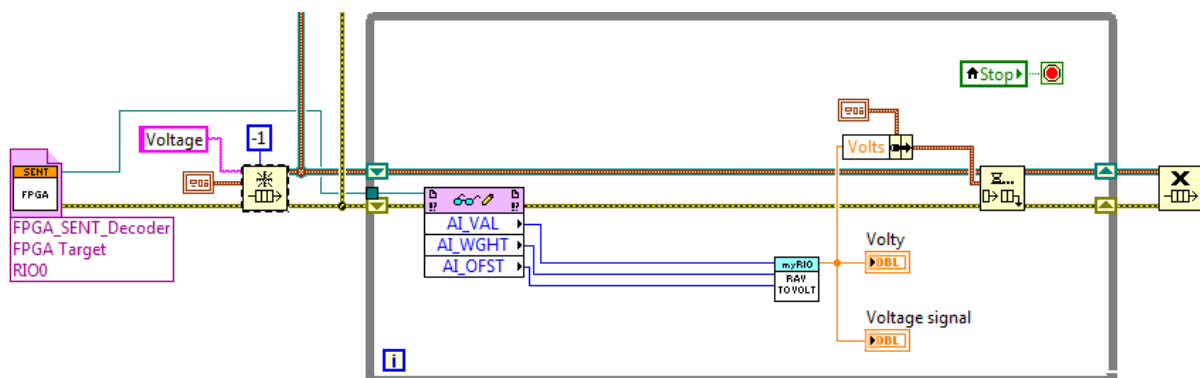
Posledním parametrem je přesnost hodin. Toto subVI vypočítává z nastaveného „SENT Tick time“  $\pm 20\%$  této hodnoty a porovnává je s rozdílem nastaveného „SENT Tick time“ a vypočítaného „Tick time“, pokud je rozdíl těchto hodnot mezi  $\pm 20\%$  je vše v pořádku, v opačném případě se rozsvítí LED zobrazující přesnost (Accuracy).



Obr. 24. SubVI pro kontrolu přesnosti hodin.

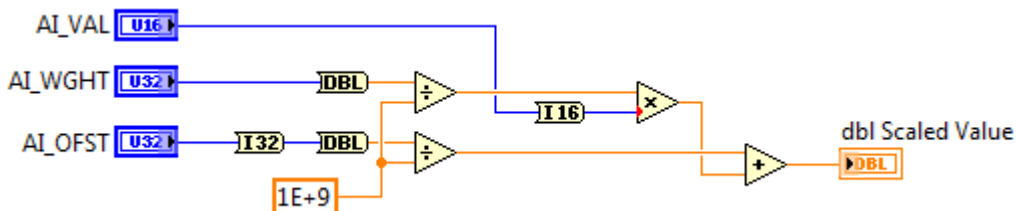
## 4.4 Napěťová oblast

Získávání analogového signálu je v aplikaci řešeno separátně od získávání signálu digitálního. Za tímto účelem byla použita nová funkce „Open FPGA VI Reference“. Dále byla vytvořena nová fronta pomocí funkce „Obtain Queue“ s názvem „Voltage“ pro přenos získaných dat do smyčky určené k jejich zpracování. Tato fronta má nastavenou velikost na minus jedna, což znamená, že její velikost není limitována.



Obr. 25. Získání hodnot analogového signálu.

Hodnota analogového signálu je poté získávána pomocí funkce „Read/Write Control“ z palety funkcí „FPGA Interface“. Tímto způsobem byla také získána hodnota váhy analogového vstupu a také hodnota analogového offsetu.



Obr. 26. SubVI pro získání hodnoty napětí.

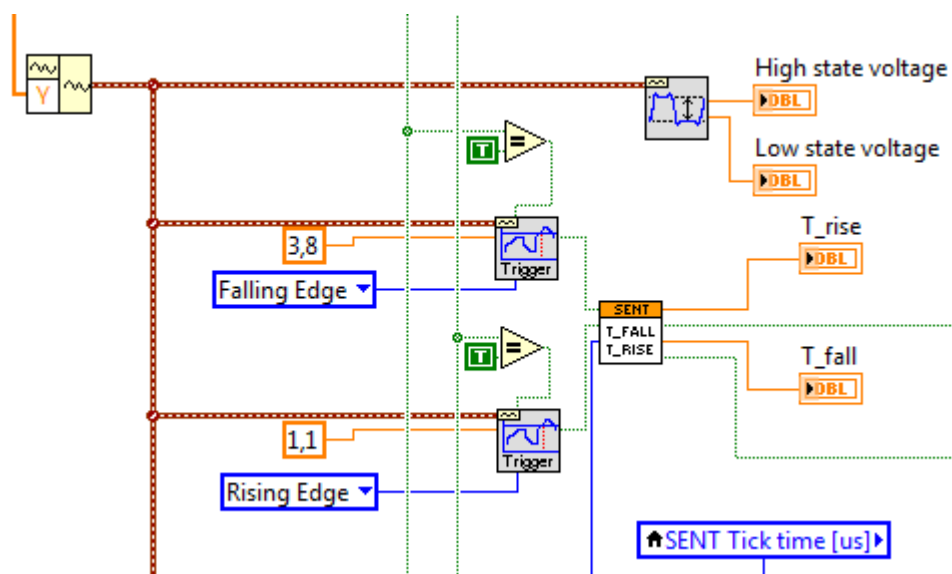
Tyto tři hodnoty jsou poté přivedeny na vstupy subVI, které přepočítává hodnotu analogového vstupu na hodnotu napětí. Hodnota je přepočtena pomocí své váhy a je k ní přičtena hodnota offsetu. Výstupem tohoto subVI jsou hodnoty napětí uvedené ve voltech. Také je zde zobrazení signálu do grafu. Tyto hodnoty jsou poté zapsány do vytvořené fronty, která přenáší do následující smyčky „While“.

V této smyčce je třeba nejdříve vytvořit pole hodnot využívající „Shift registry“ pro zapamatování předchozích hodnot. Toto pole je nutné mazat, aby program nezpomaloval svůj průběh. Maximální velikost pole je omezena na tisíc prvků.

Další krokem je signalizace „High state“ a „Low state“ stavů, která je prováděna pomocí porovnávacích funkcí. Pokud je při logické úrovni jedna hodnota napětí menší než 4,1 tak se rozsvítí dioda signalizující chybu „High state“ stavu. Podobně pokud je při logické úrovni nula hodnota napětí větší než 0,5 tak se rozsvítí dioda signalizující chybu „Low state“ stavu.

Dalším krokem bylo vytvoření „Waveform“ z připraveného pole napětíových hodnot. Tento krok je nutný kvůli použití triggrovacích funkcí. Nejprve je použita knihovní funkce „Amplitude and level“, která zobrazuje hodnotu „High state“ a „Low state“ stavu napětí.

Dále jsou použity výše zmíněné triggrovací funkce pro spouštění subVI na měření délky trvání stabilních stavů a délky trvání náběžné a sestupné hrany.

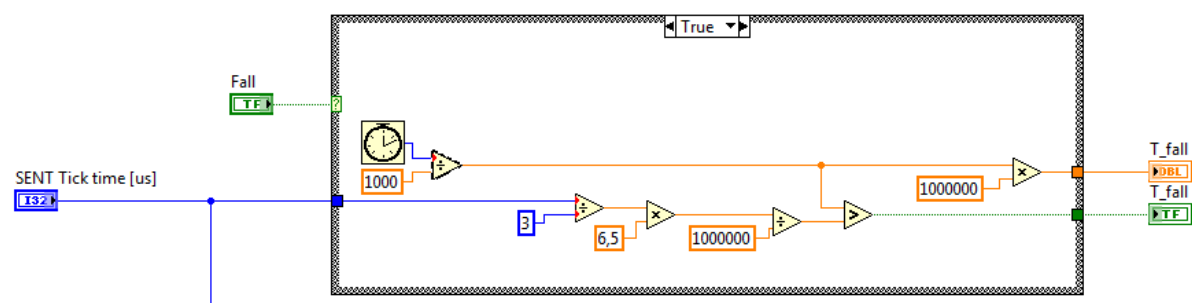


Obr. 27. Triggrování subVI.

Pro výpočet délky trvání sestupné hrany je trigger nastaven na hodnotu napětí 3,8 voltů a „Falling edge“. Pro výpočet délky trvání náběžné hrany je trigger nastaven na hodnotu napětí 1,1 voltů a „Rising edge“. Tato funkce je přivedená na vstup subVI pro měření délky trvání těchto stavů.

Triggry jsou resetovány podle „High state“ a „Low state“ stavů napětí. Jakmile je signál v „High state“ stavu resetuje se trigger pro měření času náběžné hrany a jakmile je signál v „Low state“ stavu restartuje se trigger pro měření času sestupné hrany. Stejným způsobem jsou triggrovány subVI pro měření délky trvání stabilních stavů.

SubVI pro měření délky trvání jednotlivých úseku jsou v podstatě stejné. Po spuštění struktury case triggreem začne měření času. Jelikož je tato hodnota v milisekundách tak je poté dělena tisícem. Každý z daných úseků má jinou minimální nebo maximální délku trvání. Pro sestupnou hranu je to při 3 mikrosekundovém taktu hodin 6,5 mikrosekund. Tuto hodnotu je ale třeba přepočítat podle nastavené hodnoty „SENT Tick time“.



Obr. 28. SubVI pro měření délky trvání.

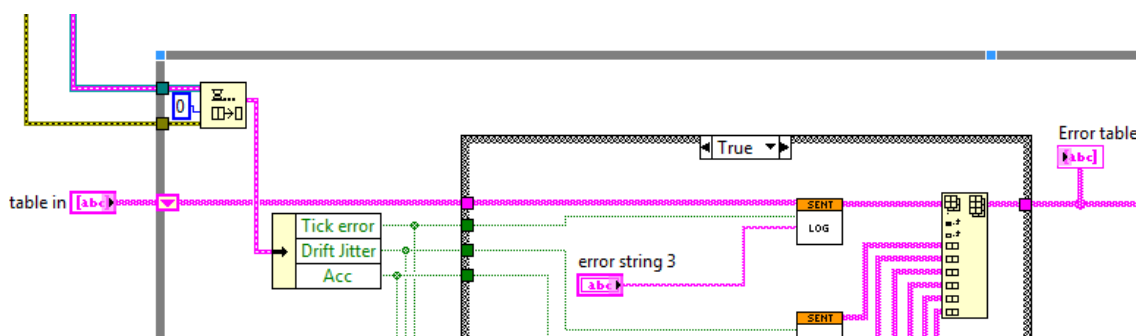
Po změření délky trvání a přepočtu se objeví výsledná hodnota na indikátoru T\_fall a v případě že je tato hodnota větší než hodnota vypočtená, rozsvítí se dioda signalizující chybu tohoto parametru.



## 4.5 Logování chybových stavů

Logování chybových stavů je prováděno pomocí dvou front. Jedna je vytvořena ve smyčce pro kontrolu časové oblasti a je pojmenovaná „Time\_log“. Inicializace této fronty se provádí před smyčkou pomocí funkce „Obtain Queue“. Velikost této fronty je nastavená na minus jeden, což opět znamená, že její velikost je neomezená.

Data jsou do fronty zapsána pomocí funkce „Lossy Enqueue Element“ a zapsanými daty jsou výstupy diod pro signalizaci chyb časové oblasti. Tyto data jsou poté přivedeny do další smyčky, která provádí jejich zápis do tabulky a poté i samotné uložení tabulky do souboru.

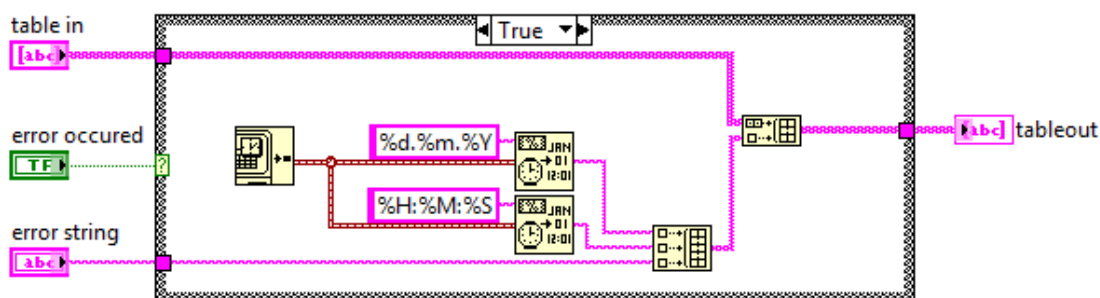


Obr. 29. Smyčka logování chybových stavů.

Druhá fronta je vytvořená před smyčkou pro zpracování parametrů napětíové oblasti. Tato fronta je pojmenovaná „U\_log“ a její velikost je také nastavena na neomezenou. Do této fronty se zapisují data indikátorů chyb napětíové oblasti. Toto je opět provedeno pomocí funkce „Lossy Enqueue Element“ a data jsou vedena do smyčky pro logování chybových stavů.

V této smyčce jsou pak vyčtená data z obou front přivedena na funkci „Compound Arithmetics“, která spouští následující strukturu case pouze v případě, že nějaký chybový stav skutečně nastal. Uvnitř této struktury se nachází subVI pro logování chybových stavů pro každý možný vzniklý chybový stav.

Vstupem do těchto subVI je chybový „string“ ve kterém je zapsaný typ chyby a také indikátor dané chyby spouštějící strukturu case nacházející se uvnitř tohoto subVI.



Obr. 30. SubVI pro logování chybových stavů.

Do subVI vstupuje již vytvořená prázdná tabulka a postupně jsou do ní vkládána data ve formě stringů. Při každém zapsání do tabulky se uloží datum, čas a typ chyby, která nastala. Datum a čas aplikace získává pomocí funkce „Get Date/Time In Seconds“. Tyto data jsou posléze pomocí funkce „Format Date/Time String“ uvedeny do správného formátu pro zápis do tabulky.

Po získání všech potřebných informací je sestaveno pomocí funkce „Build array“ pole obsahující všechny potřebné informace. Toto pole je poté zapsáno do připraveného prázdného pole tabulky, které je přivedeno na indikátor „Table Control“, který zobrazí data v tabulce.

Logování těchto chyb do souboru je prováděno pomocí struktury „Case“. Ta se spustí po stisknutí tlačítka „Table save“ a data přivedená do této smyčky jsou uložena do souboru pomocí funkce „Write To Spreadsheet File“. Je zde také nutné specifikovat cestu uloženého souboru.

## **4.6 Dekódování rychlých a pomalých zpráv**

Obsah této kapitoly podléhá utajení z důvodu ochrany firemního tajemství společnosti Continental Automotive Czech Republic s.r.o a proto je neveřejný. Plné znění této práce je uchováno na sekretariátu katedry Kybernetiky a biomedicínského inženýrství.

## **5. Verifikace řešení a testování**

Obsah této kapitoly podléhá utajení z důvodu ochrany firemního tajemství společnosti Continental Automotive Czech Republic s.r.o a proto je neveřejný. Plné znění této práce je uchováno na sekretariátu katedry Kybernetiky a biomedicínského inženýrství.

## 6. Zhodnocení

V této práci jsem se zabýval komunikací SENT, kontrolou parametrů jejich časové a napěťové oblasti a následným logováním vzniklých chybových stavů do tabulky a souboru.

Nejprve bylo nutné seznámit se se samotnou komunikací SENT, k čemuž dobře posloužila norma SAE J2716 popisující tuto komunikaci. Zde pro mě bylo celkem obtížné přeložit a správně pochopit tuto normu, ale nakonec jsem pochopil vše potřebné k vytvoření této práce. Dále bylo nutné naučit se správně pracovat se zařízením myRIO-1900, které bylo použito pro vývoj i testování této aplikace. Pro tvorbu aplikace byl použit program LabVIEW. S tímto programem jsem byl seznámen již dříve, ale jelikož bylo v tomto případě nutné využít FPGA vrstvu zařízení myRIO-1900, byl jsem nucen naučit se s touto vrstvou pracovat, což pro mě bylo něco nového.

Dále bylo nutné navrhnout strukturu celé této aplikace. Struktura byla navržena jako dvě části, jedna část pro analogový signál a druhá část pro digitální signál se společným logováním chybových stavů.

Nejtěžší částí bylo vytvoření a implementace jednotlivých částí této aplikace. Jednotlivé části aplikace byly tvořeny postupně. Nejprve byla vytvořena část pro kontrolu parametrů časové oblasti. Následně byla vytvořena část pro kontrolu parametrů napěťové oblasti a poslední částí bylo logování chybových stavů. Tato část mé práce byla vytvořena v programu LabVIEW, což je grafické vývojové prostředí pro návrh systému. Problémy při tvoření této aplikace vznikly hlavně kvůli nedostatečné znalosti funkcí programu LabVIEW a v některých případech i špatného pochopení normy SAE J2716.

Testování bylo provedeno za pomoci již zmíněného zařízení myRIO-1900 a senzoru zapůjčeného společností Continental. Samotné testování proběhlo dle očekávání. Všechny části aplikace jsou funkční.

Dalších zlepšení této práce by mohlo být několik. Jako první bych uvedl zlepšení hardware pro tuto aplikaci, jelikož maximální rychlost čtení analogového vstupu myRIO-1900 je pouze 500kS/s. Dalším zlepšením by určitě byla lepší optimalizace samotné aplikace pro zajištění větší plynulosti chodu programu.

Jelikož jsem poprvé pracoval na aplikaci výhradně podle normy, byla tato práce pro mne zprvu poněkud náročná. Také mé zkušenosti týkající se programu LabVIEW byly omezeny pouze na několik projektů. V průběhu tvorby této práce jsem se naučil lépe pracovat v programu LabVIEW a také jak správně získávat informace z normy.

## Literatura

- [1] MARTINEK, Radislav. *Senzory v průmyslové praxi*. 1. vyd. Praha: BEN - technická literatura, 2004. ISBN 80-730-0114-4.
- [2] VLACH, Jaroslav; HAVLÍČEK, Josef. *Začínáme s LabVIEW*. 1. vyd. Ilustrace Viktorie Vlachová. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [3] KREIDL, Marcel. *Technická diagnostika*. 1. vyd. Praha: BEN, 2006, 406 s. *Technická diagnostika - senzory, metody, analýza signálu*. ISBN 80-730-0158-6.
- [4] *Surface vehicle information report: SENT - Single Edge Nibble Transmission for Automotive Applications*. 2010-03. USA: SAE International, 2007-04.
- [5] *The LabVIEW RIO Architecture: A Foundation for Innovation* [online]. Austin, Texas: National Instruments, 2015 [cit. 2017-01-17]. Dostupné z: <http://www.ni.com/white-paper/10894/en/>
- [6] *FPGA Fundamentals* [online]. Austin, Texas: National Instruments, 2012 [cit. 2017-01-17]. Dostupné z: <http://www.ni.com/white-paper/6983/en/>
- [7] *USER GUIDE AND SPECIFICATIONS NI myRIO-1900* [online]. Austin, Texas: National Instruments, 2015 [cit. 2017-01-17]. Dostupné z: <http://sine.ni.com/psp/app/doc/p/id/psp-1166/lang/cs>
- [8] *Using NI CompactRIO Scan Mode with NI LabVIEW Software* [online]. Austin, Texas: National Instruments, 2015 [cit. 2017-01-17]. Dostupné z: <http://www.ni.com/white-paper/7338/en/>